

[illegible]

```
DDDDDDDD  MM      MM  DDDDDDDD  RRRRRRRR  IIIIII  VV      VV  EEEEEEEEE  RRRRRRRR
DDDDDDDD  MM      MM  DDDDDDDD  RRRRRRRR  IIIIII  VV      VV  EEEEEEEEE  RRRRRRRR
DD      DD  MMMM  MMMM  DD      DD  RR      RR  II      II  VV      VV  EE      EE  RR      RR
DD      DD  MMMM  MMMM  DD      DD  RR      RR  II      II  VV      VV  EE      EE  RR      RR
DD      DD  MM  MM  MM  DD      DD  RR      RR  II      II  VV      VV  EE      EE  RR      RR
DD      DD  MM      MM  DD      DD  RRRRRRRR  II      II  VV      VV  EE      EE  RRRRRRRR
DD      DD  MM      MM  DD      DD  RRRRRRRR  II      II  VV      VV  EEEEEEEE  RRRRRRRR
DD      DD  MM      MM  DD      DD  RR  RR      II      II  VV      VV  EE      EE  RR  RR
DD      DD  MM      MM  DD      DD  RR  RR      II      II  VV      VV  EE      EE  RR  RR
DD      DD  MM      MM  DD      DD  RR  RR      II      II  VV      VV  EE      EE  RR  RR
DD      DD  MM      MM  DDDDDDDD  RR      RR  IIIIII  VV      VV  EEEEEEEEE  RR      RR
DDDDDDDD  MM      MM  DDDDDDDD  RR      RR  IIIIII  VV      VV  EEEEEEEEE  RR      RR
DDDDDDDD  MM      MM  DDDDDDDD  RR      RR  IIIIII  VV      VV  EEEEEEEEE  RR      RR
                                     ....
                                     ....
                                     ....
                                     ....
```

```
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```

(1)	401	RK611-RK06/RK07 FUNCTION DECISION TABLE
(1)	514	TEST EVEN BYTE COUNT
(1)	551	START I/O OPERATION
(1)	997	RK611-RK06/RK07 HARDWARE FUNCTION EXECUTION
(1)	1374	RK611-RK06/RK07 CLASSIFY DRIVE TYPE AND SET PARAMETERS
(1)	1413	RK611-RK06/RK07 REGISTER DUMP ROUTINE
(1)	1448	RK06/RK07 DISK DRIVE INITIALIZATION
(1)	1501	RK611-RK06/RK07 UNSOLICITED INTERRUPT ROUTINE
(1)	1529	WAIT FOR CONTROLLER READY
(1)	1546	RK611 DISK CONTROLLER INTERRUPT DISPATCHER
(1)	1662	RK611 DISK CONTROLLER INITIALIZATION
(1)	1690	RK611 Autoconfigure Unit Delivery Routine

```
0000 1 .TITLE DMDRIVER - RK611-RK06/RK07 DISK DRIVER
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6 *****
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24
0000 25 *****
0000 26 *****
0000 27
0000 28 D. N. CUTLER 12-MAR-77
0000 29
0000 30 MODIFIED BY:
0000 31
0000 32 V03-011 RAS0300 Ron Schaefer 27-Apr-1984
0000 33 Add DEV$M_NNM characteristic to DECHAR2 so that these
0000 34 devices will have the "node$" prefix.
0000 35
0000 36 V03-010 PRD0066 Paul R. DeStefano 24-Feb-1984
0000 37 Modify DMS$INT, RETREG, and DM_UN$OLNT to compensate for
0000 38 the RK611 controller's failure to properly set/clear
0000 39 volume valid bit in drive status register.
0000 40
0000 41 V03-009 WHM0001 Bill Matthews 22-Feb-1984
0000 42 Fix a MOV$ IDB$W_UNITS(R3),R0 to be a MOV$ZWL IDB$W_UNITS(R3),R0
0000 43 in routine GET_UNITS.
0000 44
0000 45 V03-008 PRD0045 Paul R. DeStefano 11-Jan-1984
0000 46 Fix BBS instruction in DEVICE TIME OUT routine.
0000 47
0000 48 V03-007 PRD0032 Paul R. DeStefano 09-Sep-1983
0000 49 Added EXE$LCLDSKVALID to function decision table.
0000 50
0000 51 V03-006 ROW0211 Ralph O. Weber 16-AUG-1983
0000 52 Change device-dependent UCB definition base from UCB$W_BCR+2
0000 53 to UCB$K_LCL_DISK_LENGTH.
0000 54
0000 55 V03-005 PRD0024 Paul R. DeStefano 06-May-1983
0000 56 Modified RETREG routine to attempt to clear a drive
0000 57 unsafe condition.
```

0000	58	:	
0000	59	:	
0000	60	:	V03-004 PRD0019 Paul R. DeStefano 26-Apr-1983
0000	61	:	Modified FATALERR routine to return \$\$\$_PARITY only for
0000	62	:	errors that possibly indicate bad media. All other error
0000	63	:	conditions which formerly returned \$\$\$_PARITY now return
0000	64	:	\$\$\$_CNTLERR.
0000	65	:	
0000	66	:	V03-003 PRD0016 Paul R. DeStefano 26-Apr-1983
0000	67	:	Modified ECC correction logic so that ECC is only applied
0000	68	:	when there is single bit ECC correctable error, or if there
0000	69	:	is a multiple bit ECC correctable error and the error cannot
0000	70	:	be corrected using retries.
0000	71	:	
0000	72	:	V03-002 KDM0002 Kathleen D. Morse 28-Jun-1982
0000	73	:	Added \$DCDEF, \$DYNDEF, \$PRDEF, and \$\$\$DEF.
0000	74	:	
0000	75	:	V03-001 KTA0100 Kerbey T. Altmann 07-Jun-1982
0000	76	:	Add code to set UCB\$_MEDIA_ID.
0000	77	:	
0000	78	:	**

```
0000 80 : RK611-RK06/RK07 DISK DRIVER
0000 81 :
0000 82 : MACRO LIBRARY CALLS
0000 83 :
0000 84 :
0000 85 $ADPDEF ;DEFINE ADP OFFSETS
0000 86 $ACFDEF ;DEFINE ACF OFFSETS
0000 87 $CRBDEF ;DEFINE CRB OFFSETS
0000 88 $DCDEF ;DEFINE DEVICE CLASSES
0000 89 $DEVDEF ;DEFINE DEVICE CHARACTERISTICS BITS
0000 90 $DDBDEF ;DEFINE DDB OFFSETS
0000 91 $DPTDEF ;DEFINE DPT OFFSETS
0000 92 $DYNDEF ;DEFINE DYNAMIC DATA STRUCTURE TYPES
0000 93 $EMBDEF ;DEFINE EMB OFFSETS
0000 94 $IDBDEF ;DEFINE IDB OFFSETS
0000 95 $IPLDEF ;DEFINE USEFUL IPLs
0000 96 $IODEF ;DEFINE I/O FUNCTION CODES
0000 97 $IRPDEF ;DEFINE IRP OFFSETS
0000 98 $PRDEF ;DEFINE PROCESSOR REGISTERS
0000 99 $SSDEF ;DEFINE SYSTEM STATUS CODES
0000 100 $UCBDEF ;DEFINE UCB OFFSETS
0000 101 $VECDDEF ;DEFINE INTERRUPT DISPATCH VECTOR OFFSETS
0000 102 :
0000 103 :
0000 104 : LOCAL MACROS
0000 105 :
0000 106 : EXECUTE FUNCTION AND BRANCH ON RETRIABLE ERROR CONDITION
0000 107 :
0000 108 :
0000 109 .MACRO EXFUNCH BDST,FCODE
0000 110 .IF NB FCODE
0000 111 MOVZBL #CD'FCODE,R3
0000 112 .ENDC
0000 113 BSBW FEXH
0000 114 .SIGNED_BYTE BDST--1
0000 115 .ENDM
0000 116 :
0000 117 .MACRO EXFUNCL BDST,FCODE
0000 118 .IF NB FCODE
0000 119 MOVZBL #CD'FCODE,R3
0000 120 .ENDC
0000 121 BSBW FEXL
0000 122 .SIGNED_BYTE BDST--1
0000 123 .ENDM
0000 124 :
0000 125 :
0000 126 : GENERATE FUNCTION TABLE ENTRY AND CASE TABLE INDEX SYMBOL
0000 127 :
0000 128 :
0000 129 .MACRO GENF FCODE
0000 130 CD'FCODE=-FTAB/2
0000 131 .WORD FCODE!RK_CS1_M_GO!RK_CS1_M_IE
0000 132 .ENDM
0000 133 :
0000 134 :
0000 135 : LOCAL SYMBOLS
0000 136 :
```

```
0000 137 ; RK611-RK06/RK07 CONTROLLER REGISTER OFFSETS
0000 138 ;
0000 139 ;
0000 140 $DEFINI RK
0000 141
0000 142 $DEF RK_CS1 .BLKW 1 ; CONTROL STATUS REGISTER 1
0002 143 _VIELD RK_CS1,0,<- ; CONTROL STATUS REGISTER 1 FIELD DEFINITION
0002 144 <GO,,M>,- ; GO BIT
0002 145 <FCODE,,4>,- ; FUNCTION CODE
0002 146 <DPPE,,M>,- ; DATA PATH PURGE ERROR
0002 147 <IE,,M>,- ; INTERRUPT ENABLE
0002 148 <RDY,,M>,- ; CONTROLLER READY
0002 149 <MEX,,2>,- ; MEMORY EXTENSION BITS
0002 150 <CDT,,M>,- ; CONTROLLER DRIVE TYPE
0002 151 <CTO,,M>,- ; CONTROLLER TIME OUT
0002 152 <CFMT,,M>,- ; CONTROLLER FORMAT ERROR
0002 153 <SPAR,,M>,- ; SERIAL BUS PARITY ERROR
0002 154 <DI,,M>,- ; DRIVE INTERRUPT
0002 155 <CERR,,M>,- ; CONTROLLER ERROR
0002 156 >
0002 157 $DEF RK_WC .BLKW 1 ; WORD COUNT REGISTER
0004 158 $DEF RK_BA .BLKW 1 ; BUFFER ADDRESS REGISTER
0006 159 $DEF RK_DA .BLKW 1 ; DESIRED SECTOR/TRACK ADDRESS REGISTER
0008 160 _VIELD RK_DA,0,<- ; DESIRED ADDRESS FIELD DEFINITIONS
0008 161 <SA,,5>,- ; DESIRED SECTOR ADDRESS
0008 162 <,3>,- ; RESERVED BITS
0008 163 <TA,,3>,- ; DESIRED TRACK ADDRESS
0008 164 >
0008 165 $DEF RK_CS2 .BLKW 1 ; CONTROL STATUS REGISTER 2
000A 166 _VIELD RK_CS2,0,<- ; CONTROL STATUS REGISTER 2 FIELD DEFINITION
000A 167 <DS,,3>,- ; DRIVE SELECT
000A 168 <RLS,,M>,- ; RELEASE DRIVE
000A 169 <BAI,,M>,- ; BUFFER ADDRESS INCREMENT INHIBIT
000A 170 <SCLR,,M>,- ; SUBSYSTEM CLEAR
000A 171 <IR,,M>,- ; INPUT READY
000A 172 <OR,,M>,- ; OUTPUT READY
000A 173 <UFE,,M>,- ; UNIT FIELD ERROR
000A 174 <MDS,,M>,- ; MULTIPLE DRIVE SELECT
000A 175 <PGE,,M>,- ; PROGRAMMING ERROR
000A 176 <NEM,,M>,- ; NONEXISTENT MEMORY
000A 177 <NED,,M>,- ; NONEXISTENT DRIVE
000A 178 <UPE,,M>,- ; UNIBUS PARITY ERROR
000A 179 <WCE,,M>,- ; WRITE CHECK ERROR
000A 180 <DLT,,M>,- ; DATA LATE ERROR
000A 181 >
000A 182 $DEF RK_DS .BLKW 1 ; DRIVE STATUS REGISTER
000C 183 _VIELD RK_DS,0,<- ; DRIVE STATUS REGISTER BIT DEFINITIONS
000C 184 <DRA,,M>,- ; DRIVE AVAILABLE
000C 185 <,1>,- ; RESERVED BIT
000C 186 <OFST,,M>,- ; DRIVE OFFSET
000C 187 <ACLO,,M>,- ; DRIVE AC LOW
000C 188 <DCLO,,M>,- ; DRIVE DC LOW
000C 189 <DROT,,M>,- ; DRIVE OFF TRACK
000C 190 <VV,,M>,- ; VOLUME VALID
000C 191 <DRDY,,M>,- ; DRIVE READY
000C 192 <DDT,,M>,- ; DRIVE DRIVE TYPE
000C 193 <,2>,- ; RESERVED BITS
```

```
000C 194 <WRL,,M>,- ; DRIVE WRITE LOCKED
000C 195 <,1>,- ; RESERVED BIT
000C 196 <PIP,,M>,- ; POSITIONING IN PROGRESS
000C 197 <DSC,,M>,- ; DRIVE STATUS CHANGE
000C 198 <SVAL,,M>,- ; DRIVE STATUS VALID
000C 199 >
000C 200 $DEF RK_ER .BLKW 1 ; ERROR REGISTER
000E 201 _VIELD RK_ER,0,<- ; ERROR REGISTER BIT DEFINITIONS
000E 202 <ILF,,M>,- ; ILLEGAL FUNCTION
000E 203 <SKI,,M>,- ; SEEK INCOMPLETE
000E 204 <NXF,,M>,- ; NONEXECUTABLE FUNCTION
000E 205 <DRPAR,,M>,- ; DRIVE PARITY ERROR
000E 206 <FMTE,,M>,- ; FORMAT ERROR
000E 207 <DTYE,,M>,- ; DRIVE TYPE ERROR
000E 208 <ECH,,M>,- ; ECC HARD ERROR
000E 209 <BSE,,M>,- ; BAD SECTOR ERROR
000E 210 <HVRC,,M>,- ; HEADER VRC ERROR
000E 211 <COE,,M>,- ; CYLINDER OVERFLOW ERROR
000E 212 <IDAE,,M>,- ; INVALID DISK ADDRESS ERROR
000E 213 <WLE,,M>,- ; WRITE LOCK ERROR
000E 214 <DTE,,M>,- ; DRIVE TIMING ERROR
000E 215 <OPI,,M>,- ; OPERATION INCOMPLETE
000E 216 <UNS,,M>,- ; DRIVE UNSAFE
000E 217 <DCK,,M>,- ; DATA CHECK ERROR
000E 218 >
000E 219 $DEF RK_AS .BLKW 1 ; ATTENTION SUMMARY/OFFSET REGISTER
0010 220 _VIELD RK_AS,0,<- ; ATTENTION SUMMARY/OFFSET REGISTER FIELDS
0010 221 <OF,7>,- ; DRIVE OFFSET
0010 222 <,1>,- ; RESERVED BIT
0010 223 <ATTN,8,M>- ; DRIVE ATTENTION SUMMARY
0010 224 >
0010 225 $DEF RK_DC .BLKW 1 ; DESIRED CYLINDER ADDRESS
0012 226 $DEF RK_SPR .BLKW 1 ; UNUSED REGISTER
0014 227 $DEF RK_DB .BLKW 1 ; DATA BUFFER REGISTER
0016 228 $DEF RK_MR1 .BLKW 1 ; MAINTENANCE REGISTER 1
0018 229 _VIELD RK_MR1,0,<<MS,3>> ; MAINTENANCE REGISTER 1 FIELD DEFINITION
0018 230 $DEF RK_EC1 .BLKW 1 ; ECC POSITION REGISTER
001A 231 _VIELD RK_EC1,0,<<EPS,13>> ; ECC POSITION FIELD
001A 232 $DEF RK_EC2 .BLKW 1 ; ECC PATTERN REGISTER
001C 233 _VIELD RK_EC2,0,<<EPT,11>> ; ECC PATTERN FIELD
001C 234 $DEF RK_MR2 .BLKW 1 ; MAINTENANCE REGISTER 2
001E 235 $DEF RK_MR3 .BLKW 1 ; MAINTENANCE REGISTER 3
0020 236
0020 237 $DEFEND RK
0000 238
0000 239 :
0000 240 : SOFTWARE STATUS IN UPPER BYTE OF OFFSET WORD
0000 241 :
0000 242 :
0000 243 _VIELD DM,0,<- ; SOFTWARE STATUS BIT DEFINITIONS
0000 244 <ECI,,M>,- ; ECC INHIBIT
0000 245 <DCK,,M>,- ; DATACHECK IN PROGRESS
0000 246 <ECC_DEFER,,M>,- ; Flag to indicate that ECC correction
0000 247 > ; has been deferred until offset
0000 248 ; retries are exhausted.
0000 249
0000 250 ;
```

```
0000 251 : DEFINE DEVICE DEPENDENT UNIT CONTROL BLOCK OFFSETS
0000 252 :
0000 253 :
0000 254 $DEFINI UCB
0000 255
000000CC 0000 256 .=UCBSK_LCL_DISK_LENGTH ; Establish device-dependent base
00CC 257
00CC 258 $DEF UCBSW_DM_DTYP .BLKW 1 ;DRIVE TYPE MASK
00CE 259 $DEF UCBSW_DM_CS1 .BLKW 1 ;CONTROL STATUS REGISTER 1
00D0 260 $DEF UCBSW_DM_WC .BLKW 1 ;WORK COUNT REGISTER
00D2 261 $DEF UCBSW_DM_BA .BLKW 1 ;BUFFER ADDRESS REGISTER
00D4 262 $DEF UCBSW_DM_DA .BLKW 1 ;DISK ADDRESS REGISTER
00D6 263 $DEF UCBSW_DM_CS2 .BLKW 1 ;CONTROL STATUS REGISTER 2
00D8 264 $DEF UCBSW_DM_DS .BLKW 1 ;DRIVE STATUS REGISTER
00DA 265 $DEF UCBSW_DM_ER .BLKW 1 ;ERROR REGISTER
00DC 266 $DEF UCBSW_DM_AS .BLKW 1 ;ATTENTION SUMMARY REGISTER
00DE 267 $DEF UCBSW_DM_DC .BLKW 1 ;DESIRED CYLINDER REGISTER
00E0 268 $DEF UCBSW_DM_MR1 .BLKW 1 ;MAINTENANCE REGISTER 1
00E2 269 $DEF UCBSW_DM_MR2 .BLKW 1 ;MAINTENANCE REGISTER 2
00E4 270 $DEF UCBSW_DM_MR3 .BLKW 1 ;MAINTENANCE REGISTER 3
00E6 271 $DEF UCBSW_DM_DPN .BLKW 1 ;DATAPATH NUMBER
00E8 272 $DEF UCBSL_DM_DPR .BLKL 1 ;DATAPATH REGISTER
00EC 273 $DEF UCBSL_DM_FMPR .BLKL 1 ;FINAL MAP REGISTER
00F0 274 $DEF UCBSL_DM_PMPR .BLKL 1 ;PREVIOUS MAP REGISTER
00F4 275 $DEF UCBSW_DM_DB .BLKW 3 ;DATA BUFFER REGISTER
00FA 276 $DEF UCBSB_DM_IND .BLKB 1 ;SOFTWARE INDICATORS
00FB 277 _VIELD DM_IND,0,<- ;INDICATOR BIT DEFINITIONS
00FB 278 <OF,,M>- ; OFFSET FLAG
00FB 279 >
00000100 00FB 280 $DEF UCBSL_DM_FRS .BLKL 1 ;FINAL REQUEST STATUS
00FF 281 .BLKB 1 ;SPARE USED BYTE
00000100 0100 282
00000100 0100 283 UCBSK_DM_LENGTH=.
0100 284
0100 285 $DEFEND UCB
0000 286
0000 287 :
0000 288 : HARDWARE FUNCTION CODES
0000 289 :
0000 290
00000000 0000 291 F_NOP=0*2 ;NO OPERATION (SELECT DRIVE)
00000006 0000 292 F_UNLOAD=3*2 ;UNLOAD DRIVE
0000000E 0000 293 F_SEEK=7*2 ;SEEK CYLINDER
0000000A 0000 294 F_RECAL=5*2 ;RECALIBRATE
00000004 0000 295 F_DRVCLR=2*2 ;DRIVE CLEAR
00000000 0000 296 F_RELEASE=0*2 ;RELEASE DRIVE
0000000C 0000 297 F_OFFSET=6*2 ;OFFSET HEADS
0000000C 0000 298 F_RETCENTER=6*2 ;RETURN TO CENTERLINE
00000002 0000 299 F_PACKACK=1*2 ;PACK ACKNOWLEDGE
00000008 0000 300 F_STARTSPNDL=4*2 ;START SPINDLE
00000018 0000 301 F_WRITECHECK=12*2 ;WRITE CHECK DATA
00000012 0000 302 F_WRITEDATA=9*2 ;WRITE DATA
00000016 0000 303 F_WRITEHEAD=11*2 ;WRITE HEADER AND DATA
00000010 0000 304 F_READDATA=8*2 ;READ DATA
00000014 0000 305 F_READHEAD=10*2 ;READ HEADER AND DATA
00000000 0000 306 F_AVAILABLE=F_NOP ;DRIVE AVAILABLE (a NOP)
0000 307
```

```
0000 308 :: LOCAL DATA
0000 309 ::
0000 310 :: DRIVER PROLOGUE TABLE
0000 311 ::
0000 312 ::
0000 313 ::
0000 314 DPTAB - ;DEFINE DRIVER PROLOGUE TABLE
0000 315 END=DM END,- ;END OF DRIVER
0000 316 ADAPTER=UBA,- ;ADAPTER TYPE
0000 317 FLAGS=DPT$M_SVP,- ;SYSTEM PAGE TABLE ENTRY REQUIRED
0000 318 UCBSIZE=UCB$K_DM_LENGTH,- ;UCB SIZE
0000 319 DEFUNITS=8,- ;Default number of AUTOCONFIGURE units
0000 320 DELIVER=DM$DELIVER,- ;AUTOCONFIGURE units delivery routine
0000 321 NAME=DM DRIVER ;DRIVER NAME
0038 322 DPT_STORE INIT ;CONTROL BLOCK INIT VALUES
0038 323 DPT_STORE DDB,DDB$$_ACPD,L,<^A\F11> ;DEFAULT ACP NAME
003F 324 DPT_STORE DDB,DDB$$_ACPD+3,B,DDB$$_CART ;ACP CLASS
0043 325 DPT_STORE UCB,UCB$$_FIPL,B,8 ;FORK IPL
0047 326 DPT_STORE UCB,UCB$$_DEVCHAR,L,- ;DEVICE CHARACTERISTICS
0047 327 <DEV$M_FOD- ;FILES ORIENTED
0047 328 :DEV$M_DIR- ;DIRECTORY STRUCTURED
0047 329 :DEV$M_AVL- ;AVAILABLE
0047 330 :DEV$M_ELG- ;ERROR LOGGING ENABLED
0047 331 :DEV$M_SHR- ;SHAREABLE
0047 332 :DEV$M_IDV- ;INPUT DEVICE
0047 333 :DEV$M_ODV- ;OUTPUT DEVICE
0047 334 :DEV$M_RND> ;RANDOM ACCESS
004E 335 DPT_STORE UCB,UCB$$_DEVCHAR2,L,- ;DEVICE CHARACTERISTICS
004E 336 <DEV$M_NNM> ;PREFIX NAME WITH 'node$'
0055 337 DPT_STORE UCB,UCB$$_DEVCLASS,B,DDB$$_DISK ;DEVICE CLASS
0059 338 DPT_STORE UCB,UCB$$_DEVBUFSIZ,W,512 ;DEFAULT BUFFER SIZE
005E 339 DPT_STORE UCB,UCB$$_SECTORS,B,22 ;NUMBER OF SECTORS PER TRACK
0062 340 DPT_STORE UCB,UCB$$_TRACKS,B,3 ;NUMBER OF TRACKS PER CYLINDER
0066 341 DPT_STORE UCB,UCB$$_DIPL,B,21 ;DEVICE IPL
006A 342 DPT_STORE UCB,UCB$$_ERTCNT,B,8 ;ERROR RETRY COUNT
006E 343 DPT_STORE UCB,UCB$$_ERTMAX,B,8 ;MAX ERROR RETRY COUNT
0072 344 DPT_STORE REINIT ;CONTROL BLOCK RE-INIT VALUES
0072 345 DPT_STORE CRB,CRB$$_INTD+4,D,DM$$_INT ;INTERRUPT SERVICE ROUTINE ADDRESS
0077 346 DPT_STORE CRB,CRB$$_INTD+VEC$$_INITIAL,D,DM RK611_INIT ;CONTROLLER INIT
007C 347 DPT_STORE CRB,CRB$$_INTD+VEC$$_UNITINIT,D,DM RK0X_INIT ;UNIT INIT
0081 348 DPT_STORE DDB,DDB$$_DDT,D,DM$$_DDT ;DDT ADDRESS
0086 349 DPT_STORE END ;
0000 350 ::
0000 351 :: DRIVER DISPATCH TABLE
0000 352 ::
0000 353 ::
0000 354 ::
0000 355 DDTAB DM,- ;DRIVER DISPATCH TABLE
0000 356 DM_STARTIO,- ;START I/O OPERATION
0000 357 DM_UNSLNT,- ;UNSOLICITED INTERRUPT
0000 358 DM_FUNC$$_TABLE,- ;FUNCTION DECISION TABLE
0000 359 0,- ;CANCEL I/O ENTRY POINT
0000 360 DM_REGDUMP,- ;REGISTER DUMP ROUTINE
0000 361 <<2RK_MR3+2-4+8>*2>+<<3+5+1>*4>>,- ;SIZE OF DIAGNOSTIC BUFFER
0000 362 <<<RK_MR3+2-4+8>*2>+<1+4>+<EMBSL_DV_REGS$$_AV>> ;SIZE OF ERROR BUFFER
0038 363
0038 364 ;
```

```
0038 365 : HARDWARE I/O FUNCTION CODE TABLE
0038 366 :
0038 367 :
0038 368 FTAB:
0038 369 GENF F_NOP : NO OPERATION
003A 370 GENF F_UNLOAD : UNLOAD VOLUME
003C 371 GENF F_SEEK : SEEK CYLINDER
003E 372 GENF F_RECAL : RECALIBRATE
0040 373 GENF F_DRVCLR : DRIVE CLEAR
0042 374 GENF F_RELEASE : RELEASE PORT
0044 375 GENF F_OFFSET : OFFSET HEADS
0046 376 GENF F_RETCENTER : RETURN HEADS TO CENTERLINE
0048 377 GENF F_PACKACK : PACK ACKNOWLEDGE
004A 378 GENF F_STARTSPNDL : START SPINDLE
004C 379 GENF F_WRITECHECK : WRITE CHECK
004E 380 GENF F_WRITEDATA : WRITE DATA
0050 381 GENF F_READDATA : READ DATA
0052 382 GENF F_WRITEHEAD : WRITE HEADERS
0054 383 GENF F_READHEAD : READ HEADER
0056 384 GENF F_AVAILABLE : DRIVE AVAILABLE
0058 385 :
0058 386 :
0058 387 : OFFSET TABLE FOR RK611-RK06/RK07
0058 388 :
0058 389 :
0058 390 OFFTAB:
00 0058 391 .BYTE 0 : RETURN TO CENTERLINE
10 0059 392 .BYTE ^X10 : +400
90 005A 393 .BYTE ^X90 : -400
20 005B 394 .BYTE ^X20 : +800
A0 005C 395 .BYTE ^XA0 : -800
30 005D 396 .BYTE ^X30 : +1200
B0 005E 397 .BYTE ^XB0 : -1200
00 005F 398 .BYTE 0 : RETURN TO CENTERLINE
00000008 0060 399 OFFSIZ=.-OFFTAB : SIZE OF OFFSET TABLE
```

```
0060 401 .SBTTL RK611-RK06/RK07 FUNCTION DECISION TABLE
0060 402 :+
0060 403 : RK611-RK06/RK07 FUNCTION DECISION TABLE
0060 404 :-
0060 405
0060 406 DM_FUNCTABLE:
0060 407 FUNCTAB
0060 408 <NOP,-
0060 409 UNLOAD,-
0060 410 SEEK,-
0060 411 RECAL,-
0060 412 DRVCLR,-
0060 413 RELEASE,-
0060 414 OFFSET,-
0060 415 RETCENTER,-
0060 416 PACKACK,-
0060 417 AVAILABLE,-
0060 418 STARTSPNDL,-
0060 419 SENSECHAR,-
0060 420 SETCHAR,-
0060 421 SENSEMODE,-
0060 422 SETMODE,-
0060 423 WRITECHECK,-
0060 424 WRITEHEAD,-
0060 425 READHEAD,-
0060 426 READLBLK,-
0060 427 WRITELBLK,-
0060 428 READPBLK,-
0060 429 WRITEPBLK,-
0060 430 READVBLK,-
0060 431 WRITEVBLK,-
0060 432 ACCESS,-
0060 433 ACPCONTROL,-
0060 434 CREATE,-
0060 435 DEACCESS,-
0060 436 DELETE,-
0060 437 MODIFY,-
0060 438 MOUNT>
0068 439 FUNCTAB
0068 440 <NOP,-
0068 441 UNLOAD,-
0068 442 SEEK,-
0068 443 RECAL,-
0068 444 DRVCLR,-
0068 445 RELEASE,-
0068 446 OFFSET,-
0068 447 RETCENTER,-
0068 448 PACKACK,-
0068 449 AVAILABLE,-
0068 450 STARTSPNDL,-
0068 451 SENSECHAR,-
0068 452 SETCHAR,-
0068 453 SENSEMODE,-
0068 454 SETMODE,-
0068 455 ACCESS,-
0068 456 ACPCONTROL,-
0068 457 CREATE,-

:FUNCTION DECISION TABLE
:LEGAL FUNCTIONS
:NO OPERATION
:UNLOAD VOLUME
:SEEK CYLINDER
:RECALIBRATE
:DRIVE CLEAR
:RELEASE PORT
:OFFSET HEADS
:RETURN HEADS TO CENTERLINE
:PACK ACKNOWLEDGE
:DRIVE AVAILABLE
:START SPINDLE
:SENSE CHARACTERISTICS
:SET CHARACTERISTICS
:SENSE MODE
:SET MODE
:WRITE CHECK
:WRITE HEADERS
:READ HEADER
:READ LOGICAL BLOCK
:WRITE LOGICAL BLOCK
:READ PHYSICAL BLOCK
:WRITE PHYSICAL BLOCK
:READ VIRTUAL BLOCK
:WRITE VIRTUAL BLOCK
:ACCESS FILE AND/OR FIND DIRECTORY ENTRY
:ACP CONTROL FUNCTION
:CREATE FILE AND/OR CREATE DIRECTORY ENTRY
:DEACCESS FILE
:DELETE FILE AND/OR DIRECTORY ENTRY
:MODIFY FILE ATTRIBUTES
:MOUNT VOLUME
:BUFFERED I/O FUNCTIONS
:NO OPERATION
:UNLOAD VOLUME
:SEEK CYLINDER
:RECALIBRATE
:DRIVE CLEAR
:RELEASE PORT
:OFFSET HEADS
:RETURN HEADS TO CENTERLINE
:PACK ACKNOWLEDGE
:DRIVE AVAILABLE
:START SPINDLE
:SENSE CHARACTERISTICS
:SET CHARACTERISTICS
:SENSE MODE
:SET MODE
:ACCESS FILE AND/OR FIND DIRECTORY ENTRY
:ACP CONTROL FUNCTION
:CREATE FILE AND/OR CREATE DIRECTORY ENTRY
```

0068	458	DEACCESS,-	:DEACCESS FILE
0068	459	DELETE,-	:DELETE FILE AND/OR DIRECTORY ENTRY
0068	460	MODIFY,-	:MODIFY FILE ATTRIBUTES
0068	461	MOUNT>	:MOUNT VOLUME
0070	462	FUNCTAB DM BYTECNT,-	:EVEN BYTE COUNT REQUIRED FUNCTIONS
0070	463	<READHEAD,-	:READ HEADER
0070	464	READLBLK,-	:READ LOGICAL BLOCK
0070	465	READPBLK,-	:READ PHYSICAL BLOCK
0070	466	READVBLK,-	:READ VIRTUAL BLOCK
0070	467	WRITECHECK,-	:WRITE CHECK
0070	468	WRITEHEAD,-	:WRITE HEADERS
0070	469	WritelBLK,-	:WRITE LOGICAL BLOCK
0070	470	WRITEPBLK,-	:WRITE PHYSICAL BLOCK
0070	471	WRITEVBLK>	:WRITE VIRTUAL BLOCK
007C	472	FUNCTAB +ACPSREADBLK,-	:READ FUNCTIONS
007C	473	<READHEAD,-	:READ HEADER
007C	474	READLBLK,-	:READ LOGICAL BLOCK
007C	475	READPBLK,-	:READ PHYSICAL BLOCK
007C	476	READVBLK>	:READ VIRTUAL BLOCK
0088	477	FUNCTAB +ACPSWRITEBLK,-	:WRITE FUNCTIONS
0088	478	<WRITECHECK,-	:WRITE CHECK
0088	479	WRITEHEAD,-	:WRITE HEADERS
0088	480	WritelBLK,-	:WRITE LOGICAL BLOCK
0088	481	WRITEPBLK,-	:WRITE PHYSICAL BLOCK
0088	482	WRITEVBLK>	:WRITE VIRTUAL BLOCK
0094	483	FUNCTAB +ACPSACCESS,<ACCESS,CREATE>	:ACCESS AND CREATE FILE OR DIRECTORY
00A0	484	FUNCTAB +ACPSDEACCESS,<DEACCESS>	:DEACCESS FILE
00AC	485	FUNCTAB +ACPSMODIFY,-	:
00AC	486	<ACPCONTROL,-	:ACP CONTROL FUNCTION
00AC	487	DELETE,-	:DELETE FILE OR DIRECTORY ENTRY
00AC	488	MODIFY>	:MODIFY FILE ATTRIBUTES
00B8	489	FUNCTAB +ACPSMOUNT,<MOUNT>	:MOUNT VOLUME
00C4	490	FUNCTAB +EXESLCLDSKVALID,-	:LOCAL DISK VALID FUNCTIONS
00C4	491	<UNLOAD,-	:UNLOAD VOLUME
00C4	492	AVAILABLE,-	:UNIT AVAILABLE
00C4	493	PACKACK>	:PACK ACKNOWLEDGE
00D0	494	FUNCTAB +EXESZEROPARM,-	:ZERO PARAMETER FUNCTIONS
00D0	495	<NOP,-	:NO OPERATION
00D0	496	UNLOAD,-	:UNLOAD VOLUME
00D0	497	RECAL,-	:RECALIBRATE
00D0	498	DRVCLR,-	:DRIVE CLEAR
00D0	499	RELEASE,-	:RELEASE PORT
00D0	500	RETCENTER,-	:RETURN HEADS TO CENTERLINE
00D0	501	STARTSPNDL,-	:START SPINDLE
00D0	502	PACKACK,-	:PACK ACKNOWLEDGE
00D0	503	AVAILABLE>	:DRIVE AVAILABLE
00DC	504	FUNCTAB +EXESONEPARM,-	:ONE PARAMETER FUNCTIONS
00DC	505	<SEEK,-	:SEEK CYLINDER
00DC	506	OFFSET>	:OFFSET HEADS
00E8	507	FUNCTAB +EXESSENSEMODE,-	:
00E8	508	<SENSECHAR,-	:SENSE CHARACTERISTICS
00E8	509	SENSEMODE>	:SENSE MODE
00F4	510	FUNCTAB +EXESSETCHAR,-	:
00F4	511	<SETCHAR,-	:SET CHARACTERISTICS
00F4	512	SETMODE>	:SET MODE

```
0100 514 .SBTTL TEST EVEN BYTE COUNT
0100 515 :+
0100 516 : DM_BYTECNT - TEST EVEN BYTE COUNT
0100 517 :
0100 518 : THIS ROUTINE IS CALLED FROM THE FUNCTION DECISION TABLE DISPATCHER TO CHECK
0100 519 : THAT THE NUMBER OF BYTES TO BE TRANSFERED IS EVEN AS THE RK611 HAS A WORD
0100 520 : COUNT RATHER THAN A BYTE COUNT REGISTER.
0100 521 :
0100 522 : INPUTS:
0100 523 :
0100 524 : R0 = SCRATCH.
0100 525 : R1 = SCRATCH.
0100 526 : R2 = SCRATCH.
0100 527 : R3 = ADDRESS OF I/O REQUEST PACKET.
0100 528 : R4 = CURRENT PROCESS PCB ADDRESS.
0100 529 : R5 = ASSIGNED DEVICE UCB ADDRESS.
0100 530 : R6 = ADDRESS OF CCB.
0100 531 : R7 = I/O FUNCTION CODE.
0100 532 : R8 = FUNCTION DECISION TABLE DISPATCH ADDRESS.
0100 533 : R9 = SCRATCH.
0100 534 : R10 = SCRATCH.
0100 535 : R11 = SCRATCH.
0100 536 : AP = ADDRESS OF FIRST FUNCTION DEPENDENT PARAMETER.
0100 537 :
0100 538 : OUTPUTS:
0100 539 :
0100 540 : THE BUFFER BYTE COUNT IS CHECKED FOR BEING EQUAL TO 0 MOD 2. IF THE CHECK
0100 541 : FAILS, THEN THE I/O OPERATION IS TERMINATED WITH AN ERROR. ELSE A RETURN
0100 542 : TO THE FUNCTION DECISION TABLE DISPATCHER IS EXECUTED.
0100 543 :-
0100 544 :
0100 545 DM_BYTECNT:
0100 546 BBS #0,4(AP),10$ ; IF SET, ODD BYTE COUNT
0100 547 RSB ;
0100 548 10$: MOVZWL #SS$ IVBUFLEN,R0 ; SET ODD BYTE COUNT STATUS
0100 549 JMP G^EXE$ABORTIO ;
```

01 04 AC 00 E0 0100 546  
50 034C 8F 05 0105 547  
00000000 GF 17 0106 548  
010B 549

```
0111 551 .SBTTL START I/O OPERATION
0111 552 :+
0111 553 : STARTIO - START I/O OPERATION ON DEVICE UNIT
0111 554 :
0111 555 : THIS ENTRY POINT IS ENTERED TO START AN I/O OPERATION ON A DEVICE UNIT.
0111 556 :
0111 557 : INPUTS:
0111 558 :
0111 559 : R3 = ADDRESS OF I/O PACKET.
0111 560 : R5 = UCB ADDRESS OF DEVICE UNIT.
0111 561 :
0111 562 : OUTPUTS:
0111 563 :
0111 564 : FUNCTION DEPENDENT PARAMETERS ARE STORED IN THE DEVICE UCB, THE ERROR
0111 565 : RETRY COUNT IS RESET, AND THE FUNCTION IS EXECUTED. AT FUNCTION COMPLETION
0111 566 : THE OPERATION IS TERMINATED THROUGH REQUEST COMPLETE.
0111 567 :-
0111 568 :
0111 569 DM_STARTIO: : START I/O OPERATION
0080 C5 0081 C5 90 0111 570 : MOVW UCBSB_ERTMAX(R5),UCBSB_ERTCNT(R5) : INITIALIZE ERROR RETRY COUNT
009A C5 20 A3 80 0118 571 : MOVW IRPSW_FUNC(R3),UCBSW_FUNC(R5) : SAVE FUNCTION CODE AND MODIFIERS
50 38 A3 D0 011E 572 : MOVL IRPSL_MEDIA(R3),R0 : GET PARAMETER LONGWORD
00FA C5 94 0122 573 : CLRB UCBSB_DM_IND(R5) : CLEAR SOFTWARE INDICATOR BYTE
00C9 C5 8A 0126 574 : BICB #DM_M_ECC_DEFER,- : Clear flag used to signal ECC
0128 575 : UCBSW_OFFSET+1(R5) : correction deferred.
012B 576 :
012B 577 : : MOVE FUNCTION DEPENDENT PARAMETERS TO UCB
012B 578 :
012B 579 :
012B 580 :
012B 581 10$: EXTZV #IRPSV_FCODE,#IRPSS_FCODE,- : EXTRACT I/O FUNCTION CODE
51 06 00 EF 012B 582 : IRPSW_FUNC(R3),R1 :
51 20 A3 91 0131 583 : CMPB #IOS_SEEK,R1 : SEEK FUNCTION?
51 02 13 0134 584 : BEQL 20$ : IF EQL YES
51 07 91 0136 585 : CMPB #IOS_RETCENTER,R1 : RETURN HEADS TO CENTERLINE?
51 25 13 0139 586 : BEQL 30$ : IF EQL YES
51 06 91 013B 587 : CMPB #IOS_OFFSET,R1 : OFFSET FUNCTION?
00BC C5 50 D0 0140 588 : BEQL 40$ : IF EQL YES
51 19 91 0145 589 : MOVL R0,UCBSW_DA(R5) : STORE PARAMETER LONGWORD
51 0A 13 0148 590 : CMPB #IOS_STARTSPNDL,R1 : Check for IOS_STARTSPNDL
51 11 91 014A 591 : BEQL 16$ : and
51 1D 12 014D 592 : CMPB #IOS_AVAILABLE,R1 : IOS_AVAILABLE, the two function
593 : BNEQ 50$ : codes which have different internal
594 : values.
51 0F 90 014F 595 : MOVW #CDF_AVAILABLE,R1 : Map IOS_AVAILABLE to CDF_AVAILABLE.
51 18 11 0152 596 : BRB 50$ :
51 09 90 0154 597 16$: MOVW #CDF_STARTSPNDL,R1 : Map IOS_STARTSPNDL to CDF_STARTSPNDL.
51 13 11 0157 598 : BRB 50$ :
0159 599 :
0159 600 : : SEEK FUNCTION - SET CYLINDER ADDRESS
0159 601 :
0159 602 :
0159 603 :
00BE C5 50 B0 0159 604 20$: MOVW R0,UCBSW_DC(R5) : SET CYLINDER ADDRESS
015E 605 : BRB 50$ :
0160 606 :
0160 607 :
```

```
0160 608 : RETURN HEADS TO CENTERLINE FUNCTION - CLEAR OFFSET VALUE
0160 609 :
0160 610 :
50 D4 0160 611 30$: CLRL R0 ;CLEAR OFFSET VALUE
0162 612 :
0162 613 :
0162 614 : OFFSET FUNCTION - SET CURRENT OFFSET VALUE
0162 615 :
0162 616 :
00CB C5 50 90 0162 617 40$: MOVB R0,UCBSW_OFFSET(R5) ;SET OFFSET VALUE
00FA C5 01 88 0167 618 BISB #DM_IND_M_OF,UCBSB_DM_IND(R5) ;SET OFFSET FLAG
016C 619 :
016C 620 :
016C 621 : FINISH PREPROCESSING
016C 622 :
016C 623 :
0092 C5 51 90 016C 624 50$: MOVB R1,UCBSB_FEX(R5) ;SAVE FUNCTION DISPATCH INDEX
68 A5 03 AA 0171 625 BICW #UCBSM_ECC!- ;CLEAR ECC CORRECTION MADE AND,
04 2A A3 07 E1 0175 626 UCBSM_DIAGBUF,UCBSW_DEVSTS(R5) ;DIAGNOSTIC BUFFER PRESENT
68 A5 02 A8 0175 627 BBC #IRPSV_DIAGBUF,IRPSW_STS(R3),FDISPATCH ;IF CLR, NO BUFFER
017A 628 BISW #UCBSM_DIAGBUF,UCBSW_DEVSTS(R5) ;SET DIAGNOSTIC BUFFER PRESENT
017E 629 :
017E 630 :
017E 631 : CENTRAL FUNCTION DISPATCH
017E 632 :
017E 633 :
017E 634 FDISPATCH: ;FUNCTION DISPATCH
53 58 A5 D0 017E 635 MOVL UCBSL_IRP(R5),R3 ;RETRIEVE ADDRESS OF I/O PACKET
0D 2A A3 08 E0 0182 636 BBS #IRPSV_PHYSIO,IRPSW_STS(R3),10$ ;IF SET, PHYSICAL I/O FUNCTION
08 64 A5 08 E0 0187 637 BBS #UCBSV_VALID,UCBSW_STS(R5),10$ ;IF SET, VOLUME SOFTWARE VALID
50 0254 8F 3C 018C 638 MOVZWL #SS$ VOLINV,R0 ;SET VOLUME INVALID STATUS
0590 31 0191 639 BRW RESETXFR ;
0194 640 :
0194 641 :
0194 642 : UNIT IS SOFTWARE VALID OR FUNCTION IS PHYSICAL I/O
0194 643 :
0194 644 :
00CB 00C9 C5 94 0194 645 10$: CLRB UCBSW_OFFSET+1(R5) ;CLEAR ECC INHIBIT AND DATACHECK IN PROGRESS
C5 01 90 0198 646 MOVB #1,UCBSB_OFFRTC(R5) ;SET INITIAL OFFSET RETRY COUNT
00CA C5 94 019D 647 CLRB UCBSB_OFFNDX(R5) ;CLEAR INITIAL OFFSET TABLE INDEX
53 0092 C5 9A 01A1 648 MOVZBL UCBSB_FEX(R5),R3 ;GET FUNCTION DISPATCH INDEX
01A6 649 CASE R3,- ;DISPATCH TO FUNCTION HANDLING ROUTINE
01A6 650 NOP,- ;NO OPERATION
01A6 651 UNLOAD,- ;UNLOAD DRIVE
01A6 652 SEEK,- ;SEEK CYLINDER
01A6 653 RECAL,- ;RECALIBRATE
01A6 654 DRVCLR,- ;DRIVE CLEAR
01A6 655 RELEASE,- ;RELEASE PORT
01A6 656 OFFSET,- ;OFFSET HEADS
01A6 657 RETCENTER,- ;RETURN TO CENTERLINE
01A6 658 PACKACK,- ;PACK ACKNOWLEDGE
01A6 659 STARTSPNDL,- ;START SPINDLE
01A6 660 WRITECHECK,- ;WRITE CHECK DATA
01A6 661 WRITEDATA,- ;WRITE DATA
01A6 662 READDATA,- ;READ DATA
01A6 663 WRITEHEAD,- ;WRITE HEADER
01A6 664 READHEAD> ;READ HEADER
```

```
01C8 665
01C8 666 :
01C8 667 : AVAILABLE -- Mark volume not valid
01C8 668 :
01C8 669 :
64 A5 0800 8F AA 01C8 670 AVAILABLE: ;DRIVE AVAILABLE
6E 11 01C8 671 BICW #UCBSM_VALID, UCBSW_STS(R5) ;Mark volume "invalid."
01CE 672 BRB NORMAL ;Complete I/O processing.
01D0 673 :
01D0 674 :
01D0 675 : UNLOAD -- Mark volume not valid
01D0 676 :
01D0 677 :
64 A5 0800 8F AA 01D0 678 UNLOAD: ;UNLOAD DRIVE
06 11 01D0 679 BICW #UCBSM_VALID, UCBSW_STS(R5) ;Mark volume "invalid."
01D6 680 BRB EXEC_FUNCTION ;Do hardware I/O operation.
01D8 681 :
01D8 682 :
01D8 683 : PACK ACKNOWLEDGE -- Mark volume valid
01D8 684 :
01D8 685 :
64 A5 0800 8F AA 01D8 686 PACKACK: ;PACK ACKNOWLEDGE
01D8 687 BICW #UCBSM_VALID, UCBSW_STS(R5) ;Mark volume "valid."
01DE 688 BRB EXEC_FUNCTION ;Do hardware I/O operation.
01DE 689 :
01DE 690 :
01DE 691 : NO OPERATION, SEEK, RECALIBRATE, DRIVE CLEAR, RELEASE, OFFSET,
01DE 692 : RETURN TO CENTER LINE, AND START SPINDLE
01DE 693 :
01DE 694 :
01DE 695 NOP: ;NO OPERATION
01DE 696 SEEK: ;SEEK CYLINDER
01DE 697 RECAL: ;RECALIBRATE
01DE 698 DRVCLR: ;DRIVE CLEAR
01DE 699 RELEASE: ;RELEASE PORT
01DE 700 OFFSET: ;OFFSET READ HEADS
01DE 701 RETCENTER: ;RETURN TO CENTERLINE
01DE 702 STARTSPNDL: ;START SPINDLE
01DE 703 EXEC_FUNCTION:
01DE 704 EXFUNCH 10$ ;EXECUTE HOUSEKEEPING FUNCTION
SA 11 01E2 705 BRB NORMAL :
0080 31 01E4 706 10$:
01E4 707 BRW RETRY ; Use BRW since EXFUNCH only allows
01E7 708 : ; for byte offset.
01E7 709 :
01E7 710 :
01E7 711 : WRITE CHECK DATA, WRITE HEADERS, AND READ HEAD
01E7 712 :
01E7 713 :
01E7 714 WRITECHECK: ;WRITE CHECK DATA
01E7 715 WRITEHEAD: ;WRITE HEADERS
01E7 716 READHEAD: ;READ HEADER
009A C5 4000 8F AA 01E7 717 BICW #IOSM_DATACHECK,UCBSW_FUNC(R5) ;CLEAR DATA CHECK REQUEST
01EE 718 :
01EE 719 :
01EE 720 : WRITE DATA, WRITE CHECK DATA, WRITE HEADERS, AND READ HEADER
01EE 721 :
```

```
00C9 C5 01 88 01EE 722
01EE 723 WRITEDATA: ;WRITE DATA
01EE 724 BISB #DM_M_ECI,UCBSW_OFFSET+1(R5) ;SET ECC INHIBIT
01F3 725
01F3 726
01F3 727 : READ DATA, WRITE DATA, WRITE CHECK DATA, WRITE HEADERS, AND READ HEADER
01F3 728
01F3 729
01F3 730 READDATA: ;READ DATA
01F3 731 BBS #IOSV_INHSEEK,UCBSW_FUNC(R5),TRANSFR ;IF SET, NO EXPLICIT SEEK
01F9 732 EXFUNCH RETRY,F_SEEK ;SEEK DESIRED CYLINDER
0200 733
0200 734 : DATA TRANSFER
0200 735
0200 736
0200 737
0200 738 TRANSFR: ;DATA TRANSFER REQUEST CHANNEL
0200 739 MOVZBL UCBSB_FEX(R5),R3 ;GET FUNCTION DISPATCH INDEX
0205 740 EXFUNCL TRANXT ;EXECUTE TRANSFER FUNCTION
0209 741
0209 742 : DATA CHECK
0209 743
0209 744
0209 745
0209 746 DATACHECK: ;DATA CHECK
0209 747 BBC #IOSV_DATACHECK,UCBSW_FUNC(R5),NORMAL ;IF CLR, NO DATA CHECK
50 009A C5 0E E1 020F 748 MOVZWL #SS$ _BASECC,R0 ;ASSUME ECC CORRECTION WAS MADE
50 0639 8F 3C 0214 749 BBS #UCBSV_ECC,UCBSW_FUNC(R5),CHECKXT ;IF SET, ECC CORRECTION MADE
27 009A C5 00 E0 021A 750 MOVBL #DM_M_CHK!- ;SET DATA CHECK IN PROGRESS
00C9 C5 03 90 021F 751 DM_M_ECI,UCBSW_OFFSET+1(R5) ;AND INHIBIT ECC CORRECTION
00CB C5 01 90 021F 752 MOVBL #1,UCBSB_OFFRT(R5) ;SET INITIAL OFFSET RETRY COUNT
00CA C5 94 0224 753 CLRB UCBSB_OFFNDX(R5) ;CLEAR INITIAL OFFSET TABLE INDEX
52 58 A5 D0 0228 754 MOVL UCBSL_IRP(R5),R2 ;GET ADDRESS OF IRP
78 A5 2C A2 7D 022C 755 MOVQ IRP$ _SVAPTE(R2),UCBSL_SVAPTE(R5) ;RESET TRANSFER PARAMETERS
00BC C5 38 A2 D0 0231 756 MOVL IRP$ _MEDIA(R2),UCBSW_DA(R5) ;
0237 757
0237 758 : DATA CHECK RETRY
0237 759
0237 760
0237 761
0237 762 CHECKRETRY: ;DATA CHECK RETRY
0237 763 EXFUNCL TRANXT,F_WRITECHECK ;EXECUTE WRITECHECK FUNCTION
023E 764
023E 765 : SUCCESSFUL OPERATION COMPLETION
023E 766
023E 767
023E 768
023E 769 NORMAL: ;
50 01 3C 023E 770 MOVZWL S^#SS$ _NORMAL,R0 ;SET NORMAL COMPLETION STATUS
017D 31 0241 771 CHECKXT: ;
0241 772 BRW FUNCXT ;
0244 773
0244 774 : TRANSFER ENDED WITH A RETRIABLE ERROR
0244 775
0244 776
0244 777
0244 778 TRANXT: ;TRANSFER EXIT
```

```
0093 C5 0B 91 0244 779 CMPB #CDF WRITEDATA,UCBSB_CEX(R5) ;WRITE DATA FUNCTION?
      1C 13 0249 780 BEQL RETRY ;IF EQL YES
0093 C5 0D 91 024B 781 CMPB #CDF WRITEHEAD,UCBSB_CEX(R5) ;WRITE HEADER FUNCTION?
      15 13 0250 782 BEQL RETRY ;IF EQL YES
51 2820 8F B3 0252 783 BITW #RK_CS1_M_CTO!- ;CONTROLLER TIMEOUT OR,
      0E 12 0257 784 RK_CS1_M_DPPE!- ;DATAPATH PURGE ERROR OR,
      8F B3 0257 785 RK_CS1_M_SPAR,R1 ;SERIAL BUS PARITY ERROR?
52 E800 8F B3 0257 786 BNEQ RETRY ;IF NEQ YES
      8F B3 0259 787 BITW #RK_CS2_M_DLT!- ;DATA LATE OR,
      0E 12 025E 788 RK_CS2_M_OPE!- ;UNIBUS PARITY ERROR OR,
      8F B3 025E 789 RK_CS2_M_NEM!- ;NONEXISTENT MEMORY OR,
      07 12 025E 790 RK_CS2_M_WCE,R2 ;WRITE CHECK ERROR?
53 200A 8F B3 025E 791 BNEQ RETRY ;IF NEQ YES
      8F B3 0260 792 BITW #RK_ER_M_DRPAR!- ;DRIVE PARITY ERROR OR,
      03 13 0265 793 RK_ER_M_OPI!- ;OPERATION INCOMPLETE OR,
      00C5 31 0265 794 RK_ER_M_SKI,R3 ;SEEK INCOMPLETE?
      03 13 0265 795 BEQL ECC ;IF EQL NO
      00C5 31 0267 796 RETRY: BRW RETRYERR ;RETRIABLE ERROR
      00C5 31 0267 797
      026A 798
      026A 799
      026A 800 : ECC, DRIVE TIMING, OR HEADER ERROR - APPLY ECC OR PERFORM OFFSET RECOVERY
      026A 801 :
      026A 802
      026A 803 ECC:
50 00C0 C5 3C 026A 804 MOVZWL UCBSW_BCR(R5),R0 ;ECC CORRECTION
50 7E A5 A0 026F 805 ADDW UCBSW_BCNT(R5),R0 ;GET NEGATIVE NUMBER OF BYTES REMAINING
51 50 D0 0273 806 MOVL R0,R1 ;CALCULATE NUMBER OF BYTES TRANSFERED
      5F 13 0276 807 BEQL OFF ;COPY NUMBER OF BYTES TRANSFERED
      02 53 08 E0 0278 808 BBS #RK_ER_V_HVRC,R3,10$ ;IF EQL NONE - PERFORM OFFSET RECOVERY
      50 D7 027C 809 DECL R0 ;IF SET, HEADER VRC ERROR
50 01FF 8F AA 027E 810 10$: BICW #^X1FF,R0 ;SET TO TRUNCATE LAST BLOCK TRANSFERED
53 1140 8F B3 0283 811 BITW #RK_ER_M_DTE!- ;TRUNCATE RESIDUAL BYTES TRANSFERED
      0288 812 RK_ER_M_ECH!- ;DRIVE TIMING ERROR OR,
      0288 813 RK_ER_M_HVRC,R3 ;ECC HARD ERROR OR,
      4D 12 0288 814 BNEQ OFF ;HEADER VRC ERROR?
      7E 52 7D 028A 815 MOVQ R2,-(SP) ;IF NEQ YES - PERFORM OFFSET RECOVERY
52 00C6 C5 0B 00 EA 028D 816 FFS #0,#11,UCBSW_EC2(R5),R2 ; Save work registers.
      0A 52 C3 0294 817 SUBL3 R2,#10,R3 ; Find the first error bit in the ECC
      09 15 0298 819 BLEQ 20$ ; pattern.
      52 D6 029A 820 INCL R2 ; Get the number of error bits
      52 EF 029C 821 EXTZV R2,R3,UCBSW_EC2(R5),R2 ; remaining in the pattern.
      0C BA 02A3 822 POPR #^M<R3,R2> ; Branch if no other bits in pattern.
      2B 1A 02A5 823 20$: BGTRU DEFER_ECC ; Point ot next bit in pattern.
      02A5 824 ; Is there more than one error bit set?
      02A7 825 ; Restore work registers without
      02A7 826 ; affecting flags.
      02A7 827 ; If more than one error bit set, don't
      02A7 828 : APPLY_ECC - ; apply ECC correction.
      02A7 829
      02A7 830 : Apply ECC correction to correct a single bit error.
      02A7 831 :
      02A7 832
      02A7 833 APPLY_ECC:
      7E 51 D0 02A7 834 MOVL R1,-(SP) ; Save total bytes transfered, inc. ECC.
00000000'GF 16 02AA 835 JSB G^I0C$APPLYECC ; Apply ECC correction.
```

```
00000000'GF 50 8ED0 02B0 836 POPL R0 ; Retrieve transfered byte count.
00CA C5 94 02B3 837 JSB G*IOCSUPDATRANSF ; Update transfer parameters.
00CA C5 94 02B9 838 CLRB UCBSB_OFFNDX(R5) ; Reset offset table index.
7E A5 B5 02BD 839 EXFUNCH 30$,F_RETCENTER ; Return to centerline.
03 13 02C4 840 TSTW UCBSW_BCNT(R5) ; Any more to transfer?
FF34 31 02C7 841 BEQL 20$ ; If EQL no.
FF3A 31 02C9 842 BRW TRANSFR ; Transfer next segment.
0079 31 02CC 843 20$: BRW DATACHECK ; Check for write check.
0079 31 02CF 844 30$: BRW FATALERR ; Branch to fatal error routine.
0079 31 02D2 845
0079 31 02D2 846
0079 31 02D2 847 : DEFER_ECC -
0079 31 02D2 848 : Don't apply ECC correction for multiple bit errors unless the error cannot
0079 31 02D2 849 : be recovered with offset retries.
0079 31 02D2 850
0079 31 02D2 851
0079 31 02D2 852
0079 31 02D2 853
0079 31 02D2 854 DEFER_ECC:
00C9 C5 04 88 02D2 855 BISB #DM_M_ECC_DEFER,- ; Set flag to indicate that ECC
00C9 C5 04 88 02D4 856 UCBSW_OFFSET+1(R5) ; can be used if offset recovery fails.
00C9 C5 04 88 02D7 857
00C9 C5 04 88 02D7 858 : OFF - OFFSET RECOVERY
00C9 C5 04 88 02D7 859 : THIS CODE IS EXECUTED WHEN A DRIVE TIMING ERROR, HEADER VRC, OR ECC HARD
00C9 C5 04 88 02D7 860 : ERROR IS DETECTED ON A READ FUNCTION.
00C9 C5 04 88 02D7 861
00C9 C5 04 88 02D7 862
00C9 C5 04 88 02D7 863
00C9 C5 04 88 02D7 864
00C9 C5 04 88 02D7 865 OFF: ; OFFSET RECOVERY
00FA C5 01 88 02D7 866 BISB #DM_IND_M_OF,UCBSB_DM_IND(R5) ; SET OFFSET FLAG
00FA C5 50 D5 02DC 867 TSTL R0 ; ANY GOOD DATA TRANSFERED?
00FA C5 0C 13 02DE 868 BEQL 10$ ; IF EQL NO
00FA C5 0C 13 02E0 869
00FA C5 0C 13 02E0 870 :
00FA C5 0C 13 02E0 871 : THE TRANSFER ENDED IN AN ERROR BUT THERE WERE SECTORS TRANSFERED THAT
00FA C5 0C 13 02E0 872 : CONTAINED GOOD DATA. SINCE THE ERROR COULD HAVE BEEN CAUSED BY A CYLIN-
00FA C5 0C 13 02E0 873 : DER CROSSING, THE GOOD DATA IS SAVED AND THE TRANSFER IS RETRIED FROM THE
00FA C5 0C 13 02E0 874 : POINT OF ERROR.
00FA C5 0C 13 02E0 875 :
00FA C5 0C 13 02E0 876
00000000'GF 16 02E0 877 JSB G*IOCSUPDATRANSF ; UPDATE TRANSFER PARAMETERS
00CA C5 94 02E6 878 CLRB UCBSB_OFFNDX(R5) ; RESET OFFSET TABLE INDEX
00CA C5 0B 11 02EA 879 BRB 20$ ;
00CA C5 0B 11 02EC 880
00CA C5 0B 11 02EC 881 : NO GOOD DATA TRANSFERED - CHECK IF CHANGE IN OFFSET NEEDED
00CA C5 0B 11 02EC 882 :
00CA C5 0B 11 02EC 883 :
00CA C5 0B 11 02EC 884
00CB C5 97 02EC 885 10$: DECB UCBSB_OFFRTC(R5) ; Change current offset?
00CB C5 31 12 02F0 886 BNEQ 50$ ; If NEQ no.
00CB C5 02 90 02F2 887 MOVB #2,UCBSB_OFFRTC(R5) ; Set offset retry count.
00CA C5 96 02F7 888 20$: INCB UCBSB_OFFNDX(R5) ; Update offset table index.
50 00CA C5 9A 02FB 889 MOVZBL UCBSB_OFFNDX(R5),R0 ; Get next offset table index.
50 00CA C5 08 91 0300 890 CMPB #OFFSTZ,R0 ; All offsets tried?
50 00CA C5 08 12 0303 891 BNEQ 30$ ; Branch if not.
50 00CA C5 02 E4 0305 892 BBSC #DM_V_ECC_DEFER,- ; Correct the error with ECC if we can.
```

```
00C9 C5      0307 893      UCBSW_OFFSET+1(R5),-
          9C      030A 894      APPLY_ECC
          39      030B 895      OFFSETERR
00C8 C5  FD45 CF40 90 030D 896 30$:  MOVB  OFFTAB-1(R0),-
          05      0315 897      UCBSW_OFFSET(R5)
          10      0315 898      40$
00CB C5      0317 899      BNEQ  #16,UCBSB_OFFRTC(R5)
          01      031C 900 40$:  EXFUNCH FATALERR,F_OFFSET
          FED4  E0 0323 901 50$:  BBS   #DM_V_DCK,-
          FF08  31 0325 902      UCBSW_OFFSET+1(R5),60$
          31 0329 903      TRANSFR
          31 032C 904 60$:  BRW   CHECKRETRY
          032F 905
          032F 906
          032F 907      : RETRIABLE ERROR
          032F 908
          032F 909
          032F 910 RETRYERR:
          0080 C5 97 032F 911      DECB  UCBSB_ERTCNT(R5)
          16      13 0333 912      BEQL  FATALERR
53 2002 8F  B3 0335 913      BITW  #RK_ER_M_OPI!-
          07      13 033A 914      BEQL  RK_ER_M_SKI,R3
          FE38  31 033C 915      EXFUNCH FATALERR,F_RECAL
          31 0343 917 10$:  BRW   FDISPATCH
          0346 918
          0346 919      : ALL OFFSETS TRIED - RETRIEVE FINAL TRANSFER STATUS
          0346 920
          0346 921
          0346 922
          0346 923 OFFSETERR:
51 00CE C5 3C 0346 924      MOVZWL UCBSW_DM_CS1(R5),R1
          034B 925
          034B 926      : FATAL CONTROLLER/DRIVE ERROR, ERROR RETRY COUNT EXHAUSTED, ERROR RETRY
          034B 927      : INHIBITED, OR FINAL OFFSET TRIED
          034B 928
          034B 929
          034B 930
          034B 931 FATALERR:
          50 0254 8F 3C 034B 932      MOVZWL #SS$ VOLINV,R0
          6B 00D8 C5 06 E1 0350 933      BBC   #RK_DS_V_VV,UCBSW_DM_DS(R5),FUNCXT
          50 023C 8F 3C 0356 934      MOVZWL #SS$ UNSAFE,R0
          62 53 0E E0 035B 935      BBS   #RK_ER_V_UN$ ,R3,FUNCXT
          50 00BC 8F 3C 035F 936      MOVZWL #SS$ FORMAT,R0
          53 30 B3 0364 937      BITW  #RK_ER_M_DTYE!-
          0367 938      RK_ER_M_FMTE,R3
          0367 939      BNEQ  FUNCXT
          50 025C 8F 3C 0369 940      MOVZWL #SS$ WRITLCK,R0
          4F 53 0B E0 036E 941      BBS   #RK_ER_V_WLE,R3,FUNCXT
          50 0134 8F 3C 0372 942      MOVZWL #SS$ IVADDR,R0
          53 0600 8F B3 0377 943      BITW  #RK_ER_M_COE!-
          037C 944      RK_ER_M_IDAE,R3
          037C 945      BNEQ  FUNCXT
          50 008C 8F 3C 037E 946      MOVZWL #SS$ DRVERR,R0
          53 3007 8F B3 0383 947      BITW  #RK_ER_M_DTE!-
          0388 948      RK_ER_M_ILF!-
          0388 949      RK_ER_M_NXF!-
          : FATAL ERROR - SET STATUS
          : SET VOLUME INVALID STATUS
          : IF CLR, VOLUME INVALID
          : SET DRIVE UNSAFE STATUS
          : IF SET, DRIVE UNSAFE
          : SET FORMAT ERROR STATUS
          : DRIVE TYPE ERROR OR,
          : FORMAT ERROR?
          : IF NEQ YES
          : SET WRITE LOCK ERROR
          : IF SET, WRITE LOCK ERROR
          : SET INVALID DISK ADDRESS STATUS
          : CYLINDER ADDRESS OVERFLOW OR,
          : INVALID DISK ADDRESS ERROR?
          : IF NEQ YES
          : SET DRIVE ERROR STATUS
          : DRIVE TIMING ERROR OR,
          : ILLEGAL FUNCTION OR,
          : NONEXECUTABLE FUNCTION OR,
```

				0388	950	RK_ER_M_OPI!-	: OPERATION INCOMPLETE OR,
				0388	951	RK_ER_M_SKI,R3	: SEEK INCOMPLETE?
				0388	952	FUNCT	: IF NEQ YES
50	01F4	8F	12	038A	953	#SS\$ PARITY,R0	: Set parity error status.
53	81C0	8F	B3	038F	954	#RK_ER_M_BS\$!	: Bad sector error or,
				0394	955	RK_ER_M_DCK!-	: data check error or,
				0394	956	RK_ER_M_ECH!-	: ECC hard error or,
				0394	957	RK_ER_M_HVRC,R3	: header VRC error?
				0394	958	FUNCT	: If NEQ yes
50	0054	8F	12	0396	959	#SS\$ CTRLERR,R0	: Set fatal controller error status.
22	53	03	E0	039B	960	#RK_ER_V_DRPAR,R3,-	: Branch if drive parity error.
				039F	961	FUNCT	:
51	2020	8F	B3	039F	962	BITW	: DATAPATH PURGE ERROR OR,
				03A4	963	RK_CS1_M_DPPE!-	: SERIAL BUS PARITY ERROR?
				03A4	964	FUNCT	: IF NEQ YES
17	52	0D	12	03A6	965	BBS	: IF SET, UNIBUS PARITY ERROR
50	005C	8F	E0	03AA	966	MOVZWL	: SET DATA CHECK ERROR STATUS
OE	52	0E	E0	03AF	967	BBS	: IF SET, WRITE CHECK ERROR
50	01C4	8F	3C	03B3	968	MOVZWL	: SET NONEXISTENT DRIVE STATUS
OS	52	OC	E0	03B8	969	BBS	: IF SET, NONEXISTENT DRIVE
50	0054	8F	3C	03BC	970	MOVZWL	: SET CONTROLLER ERROR STATUS
				03C1	971		
				03C1	972		
				03C1	973		
				03C1	974		
				03C1	975		
				03C1	976	FUNCT:	: FUNCTION EXIT
00FB	C5	50	D0	03C1	977	MOVL	: SAVE FINAL REQUEST STATUS ACROSS RECAL
00000000	GF	16	03C6	978	JSB	G*IOC\$DIAGBUFILL	: FILL DIAGNOSTIC BUFFER IF PRESENT
0092	C5	OA	91	03CC	979	CMPB	: #CDF_WRITECHECK,UCB\$B_FEX(R5) ;DRIVE RELATED FUNCTION?
		14	1A	03D1	980	BGTRU	: IF GTRU YES
0092	C5	OF	91	03D3	981	CMPB	: #CDF_AVAILABLE, UCB\$B_FEX(R5) ;DRIVE RELATED FUNCTION?
		OD	13	03D8	982	BEQL	: IF EQL YES
53	58	A5	D0	03DA	983	MOVL	: RETRIEVE ADDRESS OF IRP
	00CO	C5	A1	03DE	984	ADDW3	: CALCULATE AND SAVE BYTES TRANSFERRED
00FD	C5	32	A3	03E2	985		: IRPSW_BCNT(R3),UCB\$L_DM_FRS+2(R5) ;...
OC	00FA	C5	E5	03E7	986	BBCC	: #DM_IND V_OF,UCB\$B_DM_IND(R5),20\$ ;if CLEAR, NOT IN OFFSET MODE
0080	C5	01	90	03ED	987	MOVB	: #1,UCB\$B_ERTCNT(R5)
				03F2	988		: Set error retry count to 1 to
				03F2	989		: prevent a timeout on the following
				03F2	990		: RECAL from decrementing the count
				03F2	991		: to a negative number and thereby
				03F2	992		: triggering a semi-infinite loop.
				03F2	993	EXFUNCH	: 20\$,F_RECAL
50	00FB	C5	D4	03F9	994	CLRL	: R1
			D0	03FB	995	MOVL	: UCB\$L_DM_FRS(R5),R0
				0400	995	REQCOM	: COMPLETE REQUEST

```
0406 997 .SBTTL RK611-RK06/RK07 HARDWARE FUNCTION EXECUTION
0406 998
0406 999 : FEXH - RK611-RK06/RK07 HARDWARE FUNCTION EXECUTION (HIGH PRIORITY)
0406 1000 : FEXL - RK611-RK06/RK07 HARDWARE FUNCTION EXECUTION (LOW PRIORITY)
0406 1001
0406 1002 : THIS ROUTINE IS CALLED VIA A BSB WITH A BYTE IMMEDIATELY FOLLOWING THAT
0406 1003 : SPECIFIES THE ADDRESS OF AN ERROR ROUTINE. ALL DATA IS ASSUMED TO HAVE BEEN
0406 1004 : SET UP IN THE UCB BEFORE THE CALL. THE APPROPRIATE PARAMETERS ARE LOADED
0406 1005 : INTO DEVICE REGISTERS AND THE FUNCTION IS INITIATED. THE RETURN ADDRESS
0406 1006 : IS STORED IN THE UCB AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE INTER-
0406 1007 : RUPT OCCURS, CONTROL IS RETURNED TO THE CALLER.
0406 1008
0406 1009 : INPUTS:
0406 1010
0406 1011 : R3 = FUNCTION TABLE DISPATCH INDEX.
0406 1012 : R4 = ADDRESS OF CONTROL STATUS REGISTER 1.
0406 1013 : R5 = DEVICE UNIT UCB ADDRESS.
0406 1014
0406 1015 : R0(SP) = RETURN ADDRESS OF CALLER.
0406 1016 : R4(SP) = RETURN ADDRESS OF CALLER'S CALLER.
0406 1017
0406 1018 : IMMEDIATELY FOLLOWING INLINE AT THE CALL SITE IS A BYTE WHICH CONTAINS
0406 1019 : A BRANCH DESTINATION TO AN ERROR RETRY ROUTINE.
0406 1020
0406 1021 : OUTPUTS:
0406 1022
0406 1023 : THERE ARE FOUR EXITS FROM THIS ROUTINE:
0406 1024
0406 1025 : 1. SPECIAL CONDITION - THIS EXIT IS TAKEN IF A POWER FAILURE OCCURS
0406 1026 : OR THE OPERATION TIMES OUT. IT IS A JUMP TO THE APPROPRIATE
0406 1027 : ERROR ROUTINE.
0406 1028
0406 1029 : 2. FATAL ERROR - THIS EXIT IS TAKEN IF A FATAL CONTROLLER OR DRIVE
0406 1030 : ERROR OCCURS OR IF ANY ERROR OCCURS AND ERROR RETRY IS
0406 1031 : INHIBITED. IT IS A JUMP TO THE FATAL ERROR EXIT ROUTINE.
0406 1032
0406 1033 : 3. RETRIABLE ERROR - THIS EXIT IS TAKEN IF A RETRIABLE CONTROLLER
0406 1034 : OR DRIVE ERROR OCCURS AND ERROR RETRY IS NOT INHIBITED.
0406 1035 : IT CONSISTS OF TAKING THE ERROR BRANCH EXIT.
0406 1036
0406 1037 : 4. SUCCESSFUL OPERATION - THIS EXIT IS TAKEN IF NO ERROR OCCURS
0406 1038 : DURING THE OPERATION. IT CONSISTS OF A RETURN INLINE.
0406 1039
0406 1040 : IN ALL CASES IF AN ERROR OCCURS, AN ATTEMPT IS MADE TO LOG THE ERROR.
0406 1041
0406 1042 : IN ALL CASES FINAL DRIVE AND CONTROLLER REGISTERS ARE RETURNED VIA
0406 1043 : THE GENERAL REGISTERS R1, R2, AND R3, AND THE UCB.
0406 1044
0406 1045 : R1 = CONTROL STATUS REGISTER 1.
0406 1046 : R2 = CONTROL STATUS REGISTER 2.
0406 1047 : R3 = ERROR REGISTER.
0406 1048
0406 1049 : UCBSW_EC1(R5) = ECC POSITION REGISTER.
0406 1050 : UCBSW_EC2(R5) = ECC PATTERN REGISTER.
0406 1051 : UCBSW_BCR(R5) = BYTE COUNT REGISTER.
0406 1052
0406 1053
```

```
0406 1054 .ENABL LSB
52 00000000'GF 9E 0406 1055 FEXH: MOVAB G^IOCSREQPCHANH,R2 ;FUNCTION EXECUTOR (HIGH PRIORITY)
07 11 0406 1056 BRB 10$ ;SET ADDRESS OF REQUEST CHANNEL ROUTINE
52 00000000'GF 9E 040F 1058 FEXL: MOVAB G^IOCSREQPCHANL,R2 ;FUNCTION EXECUTOR (LOW PRIORITY)
009C C5 8ED0 040F 1059 ;SET ADDRESS OF REQUEST CHANNEL ROUTINE
0093 C5 53 90 0416 1060 10$: POPL UCBSL_DPC(R5) ;SAVE DRIVER PC VALUE
62 16 041B 1061 MOVAB R3,UCBSB_CEX(R5) ;SAVE CASE INDEX
52 54 A5 3C 0420 1062 JSB (R2) ;REQUEST CHANNEL
0422 1063 MOVZWL UCBSW_UNIT(R5),R2 ;GET DEVICE UNIT NUMBER
0426 1064 R3,<- ;DISPATCH TO PROPER FUNCTION ROUTINE
0426 1065 IMMED,- ;NO OPERATION
0426 1066 IMMED,- ;UNLOAD VOLUME
0426 1067 POSIT,- ;SEEK CYLINDER
0426 1068 POSIT,- ;RECALIBRATE
0426 1069 IMMED,- ;DRIVE CLEAR
0426 1070 RELES,- ;RELEASE DRIVE
0426 1071 POSIT,- ;OFFSET HEADS
0426 1072 POSIT,- ;RETURN TO CENTERLINE
0426 1073 IMMED,- ;PACK ACKNOWLEDGE
0426 1074 IMMED,- ;START SPINDLE
0426 1075 >
00A1 31 043E 1076 BRW XFER ;TRANSFER FUNCTION
0441 1077 .DSABL LSB
0441 1078
0441 1079
0441 1080 : IMMEDIATE FUNCTION EXECUTION
0441 1081 :
0441 1082 : FUNCTIONS INCLUDE:
0441 1083 :
0441 1084 : NO OPERATION,
0441 1085 : UNLOAD VOLUME,
0441 1086 : DRIVE CLEAR,
0441 1087 : RELEASE PORT,
0441 1088 : PACK ACKNOWLEDGE, AND
0441 1089 : START SPINDLE.
0441 1090 :
0441 1091 : INTERRUPTS ARE LOCKED OUT, THE APPROPRIATE FUNCTION IS INITIATED WITH
0441 1092 : INTERRUPT ENABLE, AND A WAITFOR INTERRUPT AND KEEP CHANNEL IS EXECUTED.
0441 1093 :
0441 1094 : THESE FUNCTIONS ALL EXECUTE WITHIN A VERY SHORT TIME (15 US), BUT ARE
0441 1095 : VERY INFREQUENT AND THEREFORE ARE DONE WITH INTERRUPTS TO AVOID AN EXTRA
0441 1096 : SET OF REGISTER SAVE LOGIC.
0441 1097 :
0441 1098 :
0441 1099 RELES: BSW #RK_CS2_M_RLS,R2 ;RELEASE PORT
52 08 A8 0441 1100 ;SET PORT RELEASE BIT
0444 1101 IMMED: DSBINT ;IMMEDIATE FUNCTION EXECUTION
0444 1102 ;DISABLE INTERRUPTS
03 64 A5 05 E1 044A 1103 BBC #UCBSV_POWER,UCBSW_STS(R5),10$ ;IF CLR, POWER HAS NOT FAILED
008A 31 044F 1104 BRW ENBXIT ;ELSE, POWER HAS FAILED
08 A4 52 B0 0452 1105 10$: MOVW R2,RK_CS2(R4) ;SET UNIT NUMBER
64 FBDA CF43 00CC C5 A9 0456 1106 BISW3 UCBSW_DM_DTYP(R5),FTAB[R3],RK_CS1(R4) ;EXECUTE FUNCTION
045F 1107 WFIKPC RLSCHN,#2 ;WAITFOR INTERRUPT
0469 1108 IOFORK ;CREATE FORK PROCESS
046F 1109 BRW RLSCHN
01A1 31 0472 1110
```

```
0472 1111 :  
0472 1112 : POSITIONING FUNCTION EXECUTION  
0472 1113 :  
0472 1114 : FUNCTIONS INCLUDE:  
0472 1115 :  
0472 1116 : SEEK CYLINDER,  
0472 1117 : RECALIBRATE,  
0472 1118 : OFFSET HEADS, AND  
0472 1119 : RETURN HEADS TO CENTERLINE.  
0472 1120 :  
0472 1121 : THE OFFSET REGISTER AND CYLINDER ADDRESS REGISTERS ARE LOADED, INTERRUPTS  
0472 1122 : ARE LOCKED OUT, AND THE APPROPRIATE POSITIONING FUNCTION IS INITIATED  
0472 1123 : WITHOUT INTERRUPT ENABLE. THE CONTROLLER IS THEN POLLED FOR READY, DEVICE  
0472 1124 : INTERRUPTS ARE ENABLED, AND A WAITFOR INTERRUPT AND RELEASE CHANNEL IS  
0472 1125 : EXECUTED.  
0472 1126 :  
0472 1127 :  
0472 1128 POSIT: : POSITIONING FUNCTION  
0472 1129 : DISABLE INTERRUPTS  
0478 1130 DSBINT #UCBS$V_POWER,UCBS$W_STS(R5),ENBXIT :IF SET, POWER HAS FAILED  
OE A4 00C8 C5 B0 047D 1131 BBS  
10 A4 00BE C5 B0 0483 1132 MOVW UCBS$W_OFFSET(R5),RR_AS(R4) :SET OFFSET VALUE  
08 A4 52 B0 0489 1133 MOVW UCBS$W_DC(R5),RK_DC(R4) :SET DESIRED CYLINDER ADDRESS  
64 FBA3 CF43 00CC C5 A9 048D 1134 MOVW R2,RK_CS2(R4) :SET UNIT NUMBER  
03C3 30 0496 1135 BISW3 UCBS$W_DM_DTYP(R5),FTAB[R3],RK_CS1(R4) :EXECUTE FUNCTION  
0499 1136 BSBW DM_WAIT :WAIT FOR FUNCTION TO COMPLETE  
04A3 1137 WFI RCH RETREG,#6 :WAITFOR INTERRUPT  
04A3 1138 5$: :  
04A9 1139 DSBINT :DISABLE INTERRUPTS  
08 A4 54 A5 B0 04AE 1140 BBS #UCBS$V_POWER,UCBS$W_STS(R5),10$ :IF SET, POWER FAILURE  
64 01 00CC C5 A9 04B3 1141 MOVW UCBS$W_UNIT(R5),RK_CS2(R4) :SET UNIT NUMBER  
03A0 30 04B9 1142 BISW3 UCBS$W_DM_DTYP(R5),#F_NOP!1,RK_CS1(R4) :SELECT DRIVE TO GET STATUS  
0A A4 4000 8F B3 04BC 1143 BSBW DM_WAIT :WAIT FOR CONTROLLER READY  
0C 12 04C2 1144 BITW #RR_DS_M_DSC,RK_DS(R4) :OPERATION COMPLETE?  
D3 11 04C4 1145 BNEQ 10$ :BR IF YES  
04CE 1146 WFI RCH RETREG,#6 :WAITFOR INTERRUPT  
04D0 1147 BRB 5$ :  
04D3 1148 10$: ENBINT :ENABLE INTERRUPTS  
04D9 1149 20$: IOFORK :CREATE FORK PROCESS  
04DC 1150 BRW RETREG :  
04DC 1151 ENBXIT: :  
04DF 1152 BRW RLSCHN :  
04E2 1153 :  
04E2 1154 : TRANSFER FUNCTION EXECUTION  
04E2 1155 :  
04E2 1156 : FUNCTIONS INCLUDE:  
04E2 1157 :  
04E2 1158 : WRITE CHECK,  
04E2 1159 : WRITE DATA,  
04E2 1160 : WRITE HEADERS,  
04E2 1161 : READ DATA, AND  
04E2 1162 : READ HEADER.  
04E2 1163 :  
04E2 1164 :  
04E2 1165 : A UNIBUS DATAPATH IS REQUESTED FOLLOWED BY THE APPROPRIATE NUMBER OF MAP  
04E2 1166 : REGISTERS REQUIRED FOR THE TRANSFER. THE TRANSFER PARAMETERS ARE LOADED  
04E2 1167 : INTO THE DEVICE REGISTERS, INTERRUPTS ARE LOCKED OUT, THE FUNCTION IS
```

```
04E2 1168 : INITIATED, AND A WAITFOR INTERRUPT AND KEEP CHANNEL IS EXECUTED.
04E2 1169 :
04E2 1170 :
04E2 1171 XFER:
04E2 1172 :TRANSFER FUNCTION EXECUTION
04E8 1173 :REQUEST DATAPATH
04EE 1174 :REQUEST MAP REGISTERS
04F4 1175 :LOAD UNIBUS MAP REGISTERS
50 7E A5 3C 04F4 1175 MOVZWL UCBSW_BCNT(R5),R0 :GET TRANSFER BYTE COUNT
50 02 50 02 C6 04F8 1176 DIVL #2,R0 :CALCULATE TRANSFER WORD COUNT
02 A4 50 AE 0502 1178 DSBINT UCBSB_DIPL(R5) :DISABLE DEVICE INTERRUPTS
50 7C A5 3C 0506 1179 MNEGW R0,RK_WC(R4) :SET TRANSFER WORD COUNT
50 51 24 A5 DO 050A 1180 MOVZWL UCBSW_BOFF(R5),R0 :GET BYTE OFFSET IN PAGE
50 07 09 34 A1 FO 050E 1181 MOVL UCBSL_CRB(R5),R1 :GET ADDRESS OF CRB
50 34 A1 02 07 BO 0514 1182 INSV CRBSL_INTD+VECSW_MAPREG(R1),#9,#7,R0 :INSERT HIGH 7 BITS OF ADDRESS
50 50 50 08 EF 0518 1183 MOVW R0,RK_BA(R4) :SET BUFFER ADDRESS
50 50 FB11 CF43 78 051E 1184 EXTZV #7,#2,CRBSL_INTD+VECSW_MAPREG(R1),R0 :GET MEMORY EXTENSION BITS
06 A4 00BC C5 BO 0522 1185 ASHL #8,R0,R0 :SHIFT LEFT ONE BYTE
10 A4 00BE C5 BO 0528 1186 BISW FTAB[R3],R0 :MERGE FUNCTION CODE
08 A4 54 A5 BO 052E 1187 MOVW UCBSW_DA(R5),RK_DA(R4) :SET DESIRED TRACK AND SECTOR ADDRESS
06 64 A5 05 E1 052E 1187 MOVW UCBSW_DC(R5),RK_DC(R4) :SET DESIRED CYLINDER ADDRESS
64 50 00CC C5 BO 0534 1188 MOVW UCBSW_UNIT(R5),RK_CS2(R4) :SET UNIT NUMBER
00CE C5 8020 8F E8 0539 1189 SETIPL #31 :DISABLE INTERRUPTS
00D6 C5 00DA C5 B4 053C 1190 BBC #UCBSV_POWER,UCBSW_STS(R5),10$ :IF CLR, NO POWER FAILURE
00D8 C5 FFBF 8F AA 0541 1191 ENBINT :ENABLE INTERRUPTS
05 00CE C5 0F E0 0544 1192 BRW 60$
41 68 A5 01 E1 0547 1193 10$: BISW3 UCBSW_DM_DTYP(R5),R0,RK_CS1(R4) :EXECUTE FUNCTION
00E6 C5 00E8 C5 51 DO 054D 1194 WFIKPC 60$,#6 :WAITFOR INTERRUPT AND KEEP CHANNEL
00F0 C5 00F0 C5 D4 0557 1195 IOFORK :CREATE FORK PROCESS
50 00D2 C5 07 09 EF 055D 1196 PURDPR :PURGE DATAPATH, CHECK/CLEAR ERRORS
50 02 07 00CF C5 FO 0563 1197 BLBS R0,20$ :BRANCH IF NO DATAPATH ERROR
50 50 01EF 8F B1 0566 1198 BISW #RK_CS1_M_CERR!- :SET CONTROLLER ERROR AND
50 50 01EF 8F 18 056D 1199 RK_CS1_M_DPPE,UCBSW_DM_CS1(R5) :DATAPATH PURGE ERROR
00EC C5 6240 DO 056D 1200 CLRW UCBSW_DM_CS2(R5) :CLEAR CONTROL STATUS REGISTER 2
00F0 C5 00F0 C5 D4 0571 1201 CLRW UCBSW_DM_ER(R5) :CLEAR ERROR REGISTER
50 34 A3 00E6 C5 51 DO 0575 1202 BICW #^C<RK_DS_M_VV>,UCBSW_DM_DS(R5) :CLEAR ALL BUT VOLUME VALID
00C0 C5 00D0 C5 02 A5 057C 1203 20$: BBS #RK_CST_V_CERR,UCBSW_DM_CS1(R5),30$ :IF SET, DEVICE ERRORS
2D 00CE C5 0F E0 0582 1204 30$: BBC #UCBSV_DIAGBUF,UCBSW_DEVSTS(R5),40$ :IF CLR, NO DIAGNOSTIC BUFFER
0587 1205 30$: EXTZV #VECSV_DATAPATH,- :EXTRACT DATAPATH #
0589 1206 : FROM CRB AND SAVE IT
058A 1207 CRBSL_INTD+VECSB_DATAPATH(R3),-
058C 1208 UCBSW_DM_DPN(R5)
50 00D2 C5 07 09 EF 058F 1209 MOVL R1,UCBSL_DM_DPR(R5) :SAVE DATAPATH REGISTER CONTENTS
50 02 07 00CF C5 FO 0594 1210 EXTZV #9,#7,UCBSW_DM_BA(R5),R0 :GET LOW BITS OF FINAL
50 50 01EF 8F B1 059B 1211 : MAP REGISTER NUMBER
50 50 01EF 8F 18 059B 1212 INSV UCBSW_DM_CS1+1(R5),#7,#2,R0 :INSERT HIGH BITS OF FINAL MAP REGISTER
50 50 01EF 8F 3C 05A2 1213 CMPW #495,R0 :LEGAL MAP REGISTER NUMBER?
50 50 01EF 8F 18 05A7 1214 BGEQ 35$ :IF GEQ YES
00EC C5 6240 DO 05A7 1215 MOVZWL #495,R0 :RESTRICT MAP REGISTER NUMBER
00F0 C5 00F0 C5 D4 05AE 1216 35$: MOVL (R2)[R0],UCBSL_DM_FMPR(R5) :SAVE FINAL MAP REGISTER
50 34 A3 00E6 C5 51 DO 05B4 1217 CLRL UCBSL_DM_PMPR(R5) :CLEAR PREVIOUS MAP REGISTER CONTENTS
50 02 07 00CF C5 FO 05B8 1218 DECL R0 :CALCULATE PREVIOUS MAP REGISTER NUMBER
50 50 01EF 8F 3C 05BA 1219 CMPV #VECSV_MAPREG,#VECSS_MAPREG,- :ANY PREVIOUS MAP REGISTER?
00C0 C5 00D0 C5 02 A5 05BD 1220 CRBSL_INTD+VECSW_MAPREG(R3),R0 :
00F0 C5 6240 DO 05C0 1221 BGTR 40$ :IF GTR NO
2D 00CE C5 0F E0 05C2 1222 MOVL (R2)[R0],UCBSL_DM_PMPR(R5) :SAVE PREVIOUS MAP REGISTER
05C8 1223 40$: MULW3 #2,UCBSW_DM_WCTR5,UCBSW_BCR(R5) :CONVERT WD TO BYTE COUNT
05D0 1224 BBS #RK_CS1_V_CERR,UCBSW_DM_CS1(R5),60$ :IF SET, DEVICE ERRORS
```

```
0093 C5 0E 91 05D6 1225 CMPB #CDF_READHEAD,UCBSB_CEX(R5) ;READ HEADER FUNCTION?
      26 12 05DB 1226 BNEQ 60$ ;IF NEQ NO
      78 A5 DD 05DD 1227 PUSHL UCBSL_SVAPTE(R5) ;SAVE ADDRESS OF PTE
51 00F4 C5 9E 05E0 1228 MOVAB UCBSW_DM_DB(R5),R1 ;SET ADDRESS OF INTERNAL BUFFER
      52 06 D0 05E5 1229 MOVL #6,R2 ;SET NUMBER OF BYTES TO MOVE
      7E A5 52 B1 05E8 1230 CMPW R2,UCBSW_BCNT(R5) ;ROOM FOR FULL HEADER?
      04 1F 05EC 1231 BLSSU 50$ ;IF LSSU YES
00C0 C5 52 7E A5 3C 05EE 1232 MOVZWL UCBSW_BCNT(R5),R2 ;SET LENGTH OF PARTIAL HEADER
      52 7E A5 A3 05F2 1233 50$: SUBW3 UCBSW_BCNT(R5),R2,UCBSW_BCR(R5) ;CALCULATE TRANSFER BYTE COUNT
      00000000 GF 16 05F9 1234 JSB G*IOC$MOVTOUSER ;MOVE HEADER TO USER BUFFER
      78 A5 8ED0 05FF 1235 60$: POPL UCBSL_SVAPTE(R5) ;RESTORE ADDRESS OF PTE
      0603 1236 SETIPL UCBSB_FIPL(R5) ;INSURE PROPER IPL FOR RELEASE
      0607 1237 RELDPR ;RELEASE DATA PATH
      060D 1238 RELMPR ;RELEASE MAP REGISTERS
      0613 1239 RLSCHN: RELCHAN ;RELEASE CHANNEL
      0613 1240
      0619 1241
      0619 1242
      0619 1243 : RETURN REGISTERS
      0619 1244
      0619 1245
      0619 1246
      0619 1247 RETREG: .ENABL LSB ;RETURN FINAL DEVICE REGISTERS
51 00CE C5 3C 0619 1248 MOVZWL UCBSW_DM_CS1(R5),R1 ;RETRIEVE CONTROL STATUS REGISTER 1
52 00D6 C5 3C 061E 1249 MOVZWL UCBSW_DM_CS2(R5),R2 ;RETRIEVE CONTROL STATUS REGISTER 2
53 00DA C5 3C 0623 1250 MOVZWL UCBSW_DM_ER(R5),R3 ;RETRIEVE ERROR REGISTER
64 A5 0060 8F B3 0628 1251 BITW #UCBSM_POWER!- ;POWER FAIL OR DEVICE TIMEOUT?
      062E 1252 UCBSM_TIMEOUT,UCBSW_STS(R5) ;
      062E 1253 BNEQ 40$ ;IF NEQ YES - SPECIAL CONDITION
      1D 51 0F E0 0630 1254 BBS #RK_CS1_V_CERR,R1,5$ ;IF SET, ERROR OCCURED
      08 91 0634 1255 CMPB #CDF_PACKACK,- ;DID WE EXECUTE A PACK ACKNOWLEDGE
      0093 C5 0636 1256 UCBSB_CEX(R5) ;FUNCTION?
      51 12 0639 1257 BNEQ 30$ ;BRANCH IF NOT.
      0080 8F B3 063B 1258 BITW #RK_DS_M_DRDY,- ;DRIVE READY BIT SET?
      00D8 C5 063F 1259 UCBSW_DM_DS(R5) ;
      48 12 0642 1260 BNEQ 30$ ;BRANCH IF SO.
      0040 8F AA 0644 1261 BICW #RK_DS_M_VV,- ;FORCE VOLUME VALID BIT TO REFLECT TRUE
      00D8 C5 0648 1262 UCBSW_DM_DS(R5) ;STATUS.
64 A5 0800 8F AA 064B 1263 BICW #UCBSM_VALID,UCBSW_STS(R5) ;MARK VOLUME "INVALID"
      0093 C5 0A 91 0651 1264 5$: CMPB #CDF_WRITECHECK,UCBSB_CEX(R5) ;DRIVE RELATED FUNCTION?
      06 1B 0656 1265 BLEQU 10$ ;IF LEQU NO
00C0 C5 7E A5 AE 0658 1266 10$: MNEGW UCBSW_BCNT(R5),UCBSW_BCR(R5) ;RESET BYTE COUNT - NO TRANSFER
      065E 1267
      065E 1268 JSB G*ERL$DEVICERR ;ALLOCATE AND FILL ERROR MESSAGE BUFFER
6B 009A C5 0F E0 0664 1269 BBS #IOSV_INHRETRY,UCBSW_FUNC(R5),70$ ;IF SET, RETRY INHIBITED
      52 1700 8F B3 066A 1270 BITW #RK_CS2_M_MDS!- ;MULTIPLE DRIVE SELECT OR,
      066F 1271 RK_CS2_M_NED!- ;NONEXISTENT DISK OR,
      066F 1272 RK_CS2_M_PGE!- ;PROGRAMMING ERROR OR,
      066F 1273 RK_CS2_M_UFE,R2 ;UNIT FIELD ERROR?
      066F 1274 BNEQ 70$ ;IF NEQ YES
      53 0EB5 8F B3 0671 1275 BITW #RK_ER_M_BSE!- ;BAD SECTOR ERROR OR,
      0676 1276 RK_ER_M_COE!- ;CYLINDER ADDRESS OVERFLOW OR,
      0676 1277 RK_ER_M_DTYE!- ;DRIVE TYPE ERROR OR,
      0676 1278 RK_ER_M_FMTE!- ;FORMAT ERROR OR,
      0676 1279 RK_ER_M_IDAE!- ;INVALID DISK ADDRESS ERROR OR,
      0676 1280 RK_ER_M_ILF!- ;ILLEGAL FUNCTION OR,
      0676 1281 RK_ER_M_NXF!- ;NONEXECUTABLE FUNCTION OR,
```

```

      0676 1282      RK_ER_M_WLE,R3      ;WRITE LOCK ERROR?
      0676 1283      BNEQ 70$           ;IF NEQ YES
      1A 53 5D 12 0678 1284      BBS #RK_ER_V_UN$ ,R3,50$      ;Branch if drive is unsafe.
53 00D8 C5 06 E1 067C 1285      BBC #RK_DS_V_VV,UCB$W_DM_DS(R5),70$ ;IF CLR, VOLUME INVALID
      0682 1286
      0682 1287      ;
      0682 1288      ; RETRIABLE ERROR EXIT
      0682 1289      ;
      0682 1290
      7E 009C D5 98 0682 1291 20$: CVTBL @UCB$DPC(R5),-(SP)      ;GET BRANCH DISPLACEMENT
      009C C5 8E C0 0687 1292      ADDL (SP)+,UCB$DPC(R5)      ;CALCULATE RETURN ADDRESS - 1
      009C C5 D6 068C 1293 30$: INCL UCB$DPC(R5)      ;ADJUST TO CORRECT RETURN ADDRESS
      009C D5 17 0690 1294      JMP @UCB$C_DPC(R5)      ;RETURN TO DRIVER
      42 11 0694 1295
      0694 1296 40$: BRB 80$      ;
      0696 1297
      0696 1298      ; Check for unsafe condition and attempt to clear it.
      0696 1299      ;
      0696 1300      ;
      0696 1301
      0696 1302 50$: DSBINT      ; Disable interrupts.
      03 64 05 E1 069C 1303      BBC #UCB$V_POWER,-      ; Branch if no power failure occurred.
      FE38 31 069E 1304      BRW UCB$W_STS(R5),60$      ;
      00CC C5 A9 06A4 1305      BRW ENBXIT      ; Otherwise, enable interrupts and
      64 05 06A4 1306      ; go process error.
      06A4 1307 60$: BISW3 UCB$W_DM_DTYP(R5),-      ; Attempt to clear unsafe condition.
      06A8 1308      #F_DRVCLR!1,RR_CS1(R4)      ;
      06AA 1309      TIMEWAIT -      ; Wait for ten microseconds or until
      06AA 1310      TIME = #1,-      ; unsafe condition clears.
      06AA 1311      BITVAL = #RK_CS1 M_CERR,-      ;
      06AA 1312      SOURCE = RK_CS1(R4),-      ;
      06AA 1313      CONTEXT = W,-      ;
      06AA 1314      SENSE = .FALSE.      ;
      AD 50 E8 06CF 1315      ENBINT      ; Enable interrupts.
      06D2 1316      BLBS R0,20$      ; Branch if drive is no longer unsafe.
      06D5 1317
      06D5 1318      ; FATAL CONTROLLER OR DRIVE ERROR EXIT
      06D5 1319      ;
      06D5 1320      ;
      06D5 1321
      FC73 31 06D5 1322 70$: BRW FATALERR      ;
      06D8 1323
      06D8 1324      ; SPECIAL CONDITION (POWER FAILURE OR DEVICE TIME OUT)
      06D8 1325      ;
      06D8 1326      ;
      06D8 1327
      54 64 A5 05 E4 06D8 1328 80$: BBSC #UCB$V_POWER,UCB$W_STS(R5),110$ ;IF SET, POWER FAILURE
      06DD 1329
      06DD 1330      ; DEVICE TIME OUT
      06DD 1331      ;
      06DD 1332      ;
      06DD 1333
      00000000'GF 16 06DD 1334      JSB G^ERL$DEVICTMO      ;LOG DEVICE TIME OUT
      53 24 A5 D0 06E3 1335      MOVL UCB$DPC(R5),R3      ;GET ADDRESS OF CRB
      53 2C A3 D0 06E7 1336      MOVL CRB$DPC+VE$C$IDB(R3),R3 ;GET ADDRESS OF IDB
      04 A3 55 D1 06EB 1337      CMPL R5,IDB$DPC_OWNER(R3) ;DEVICE OWN CONTROLLER?
      06EF 1338      BNEQ 90$      ;IF NEQ NO
```

```
08 A4 20 B0 06F1 1339      MOVW  #RK_CS2_M_SCLR,RK_CS2(R4) ;CLEAR ENTIRE RK611 SUBSYSTEM
64 40 8F 9B 06F5 1340      MOVZBW #RK_CS1_M_IE,RK_CS1(R4) ;ENABLE DEVICE INTERRUPTS
                                06F9 1341 90$: SETIPL UCB$B_FIPC(R5) ;LOWER TO FORK LEVEL
                                06FD 1342
                                06FD 1343      BBS  #RK_DS_V_DRDY,-
07 00D8 C5 E0 06FF 1344      UCB$W_DM-DS(R5),100$ ;BR. IF DEVICE READY
50 01A4 8F 3C 0703 1345      MOVZWL #SS$_MEDOFFL,R0 ;RETURN MEDIUM OFFLINE ERROR
    1A 11 0708 1346      BRB  RESETXFR ;EXIT WITHOUT RETRY
    070A 1347
50 022C 8F 3C 070A 1348 100$: MOVZWL #SS$_TIMEOUT,R0 ;SET DEVICE TIMEOUT STATUS
    0080 C5 97 070F 1349      DECB  UCB$B_ERTCNT(R5) ;ANY ERROR RETRIES REMAINING?
    OF 13 0713 1350      BEQL  RESETXFR ;IF EQL NO
64 A5 0040 8F AA 0715 1351      RELCHAN ;RELEASE CHANNEL IF OWNED
    FASA 31 071B 1352      BICW  #UCB$M_TIMEOUT,UCB$W_STS(R5) ;CLEAR TIME OUT STATUS
    0721 1353      BRW  FDISPATCH ;
    0724 1354
    0724 1355 ;
    0724 1356 ; RESET TRANSFER BYTE COUNT TO ZERO
    0724 1357 ;
    0724 1358
    0724 1359 RESETXFR:
00C0 53 58 A5 D0 0724 1360      MOVL  UCB$L_IRP(R5),R3 ;RETRIEVE ADDRESS OF I/O PACKET
    C5 32 A3 AE 0728 1361      MNEGW IRP$W_BCNT(R3),UCB$W_BCR(R5) ;RESET TRANSFER BYTE COUNT
    FC90 31 072E 1362      BRW  FUNCXT ;
    0731 1363
    0731 1364 ;
    0731 1365 ; POWER FAILURE
    0731 1366 ;
    0731 1367
    0731 1368 110$: RELCHAN ;RELEASE CHANNEL IF OWNED
78 A5 58 A5 D0 0737 1369      MOVL  UCB$L_IRP(R5),R3 ;RETRIEVE ADDRESS OF I/O PACKET
    2C A3 7D 073B 1370      MOVQ  IRP$L_SVAPTE(R3),UCB$L_SVAPTE(R5) ;RESTORE TRANSFER PARAMETERS
    F9CE 31 0740 1371      BRW  DM_STARTIO ;
    0743 1372      .DSABL LSB ;
```

```
0743 1374 .SBTTL RK611-RK06/RK07 CLASSIFY DRIVE TYPE AND SET PARAMETERS
0743 1375 :
0743 1376 : DM_DTYPE - RK611-RK06/RK07 CLASSIFY DRIVE TYPE AND SET PARAMETERS
0743 1377 :
0743 1378 : THIS ROUTINE IS CALLED WHEN AN UNSOLICITED INTERRUPT OCCURS ON A DRIVE, DURING
0743 1379 : SYSTEM INITIALIZATION, AND AT POWER RECOVERY TO DETERMINE THE DRIVE TYPE AND
0743 1380 : SET UNIT PARAMETERS.
0743 1381 :
0743 1382 : INPUTS:
0743 1383 :
0743 1384 : R4 = ADDRESS OF CONTROL STATUS REGISTER 1.
0743 1385 : R5 = DEVICE UNIT UCB ADDRESS.
0743 1386 :
0743 1387 : OUTPUTS:
0743 1388 :
0743 1389 : THE DRIVE STATUS REGISTER IS INTERROGATED AND UNIT PARAMETERS ARE SET.
0743 1390 :
0743 1391 :
0743 1392 DM_DTYPE: :CLASSIFY DRIVE TYPE AND SET UNIT PARAMETERS
0743 1393 :
0743 1394 : MOV B S^#DTS_RK06,- :SET RK06 DEVICE TYPE
0743 1395 : UCBSB_DEVTYPE(R5) :SET NUMBER OF RK06 CYLINDERS
0743 1396 : #411,UCBSW_CYLINDERS(R5);
0743 1397 : MOVZWL #411*3*22,=
0743 1398 : UCBSL_MAXBLOCK(R5) :SET MAXIMUM BLOCK NUMBER
0743 1399 : #^X2364B006,-
0743 1400 : UCBSL_MEDIA_ID(R5) :SET MEDIA IDENT 'DM RK06'
0743 1401 : UCBSW_DM_DTYP(R5) :SET RK06 CONTROLLER DRIVE TYPE
0743 1402 : #RK_DS_M-DDT,RK_DS(R4) :RK06 DRIVE?
0743 1403 : 10$ :IF EQL YES
0743 1404 : MOV B S^#DTS_RK07,- :SET RK07 DEVICE TYPE
0743 1405 : UCBSB_DEVTYPE(R5) :SET NUMBER OF RK07 CYLINDERS
0743 1406 : #815,UCBSW_CYLINDERS(R5);
0743 1407 : MOVZWL #815*3*22,=
0743 1408 : UCBSL_MAXBLOCK(R5) :SET MAXIMUM BLOCK NUMBER
0743 1409 : UCBSL_MEDIA_ID(R5) :SET MEDIA IDENT 'DM RK07'
0743 1410 : #RK_CS1_M_CDT,-
0743 1411 : UCBSW_DM_DTYP(R5) :SET RK07 CONTROLLER DRIVE TYPE
10$: RSB
```

46 A5	019B 8F	B0	0747 1395	MOVW	#411,UCBSW_CYLINDERS(R5);	SET NUMBER OF RK06 CYLINDERS
	69F6 8F	3C	074D 1396	MOVZWL	#411*3*22,=	
	00B0 C5		0751 1397		UCBSL_MAXBLOCK(R5)	SET MAXIMUM BLOCK NUMBER
2364B006 8F	D0	0754 1398	075A 1399	MOVL	#^X2364B006,-	
008C C5		075D 1400	0761 1401	CLRW	UCBSW_DM_DTYP(R5)	SET RK06 CONTROLLER DRIVE TYPE
0A A4	0100 8F	B3	0767 1402	BITW	#RK_DS_M-DDT,RK_DS(R4)	RK06 DRIVE?
	1C	13	0769 1403	BEQL	10\$	IF EQL YES
	02	90	076B 1404	MOVB	S^#DTS_RK07,-	SET RK07 DEVICE TYPE
46 A5	032F 8F	B0	076D 1405	MOVW	#815,UCBSW_CYLINDERS(R5);	SET NUMBER OF RK07 CYLINDERS
	D21E 8F	3C	0773 1406	MOVZWL	#815*3*22,=	
	00B0 C5		0777 1407		UCBSL_MAXBLOCK(R5)	SET MAXIMUM BLOCK NUMBER
	008C C5	D6	077A 1408	INCL	UCBSL_MEDIA_ID(R5)	SET MEDIA IDENT 'DM RK07'
	0400 8F	A8	077E 1409	BISW	#RK_CS1_M_CDT,-	
	00CC C5		0782 1410		UCBSW_DM_DTYP(R5)	SET RK07 CONTROLLER DRIVE TYPE
		05	0785 1411	RSB		

```
0786 1413 .SBTTL RK611-RK06/RK07 REGISTER DUMP ROUTINE
0786 1414 :
0786 1415 : DM_REGDUMP - RK611-RK06/RK07 REGISTER DUMP ROUTINE
0786 1416 :
0786 1417 : THIS ROUTINE IS CALLED TO SAVE THE CONTROLLER AND DRIVE REGISTERS IN A
0786 1418 : SPECIFIED BUFFER. IT IS CALLED FROM THE DEVICE ERROR LOGGING ROUTINE AND
0786 1419 : FROM THE DIAGNOSTIC BUFFER FILL ROUTINE.
0786 1420 :
0786 1421 : INPUTS:
0786 1422 :
0786 1423 : R0 = ADDRESS OF REGISTER SAVE BUFFER.
0786 1424 : R4 = ADDRESS OF CONTROL STATUS REGISTER 1.
0786 1425 : R5 = DEVICE UNIT UCB ADDRESS.
0786 1426 :
0786 1427 : OUTPUTS:
0786 1428 :
0786 1429 : THE CONTROLLER AND DRIVE REGISTERS ARE SAVED IN THE SPECIFIED BUFFER.
0786 1430 :
0786 1431 :
0786 1432 DM_REGDUMP:
0786 1433 :RK611-RK06/RK07 REGISTER DUMP ROUTINE
51 80 12 D0 0786 1433 MOVL #<RK MR3+2-4+8>/2,(R0)+ ;INSERT NUMBER OF DEVICE REGISTERS
00CE C5 DE 0789 1434 MOVAL UCBSW_DM_CS1(R5),R1 ;GET ADDRESS OF SAVED DEVICE REGISTERS
52 0A D0 078E 1435 MOVL #<RK MR1+2-4>/2,R2 ;SET NUMBER OF REGISTERS TO MOVE
80 81 3C 0791 1436 10$: MOVZWL (R1)+,(R0)+ ;MOVE REGISTER TO BUFFER
FA 52 F5 0794 1437 SOBGTR R2,10$ ;ANY MORE TO MOVE?
80 00C4 C5 3C 0797 1438 MOVZWL UCBSW_EC1(R5),(R0)+ ;INSERT ECC POSITION REGISTER
80 00C6 C5 3C 079C 1439 MOVZWL UCBSW_EC2(R5),(R0)+ ;INSERT ECC PATTERN REGISTER
80 81 3C 07A1 1440 MOVZWL (R1)+,(R0)+ ;INSERT MAINTENANCE REGISTER 2
80 81 3C 07A4 1441 MOVZWL (R1)+,(R0)+ ;INSERT MAINTENANCE REGISTER 3
80 81 3C 07A7 1442 MOVZWL (R1)+,(R0)+ ;INSERT DATAPATH NUMBER
80 81 D0 07AA 1443 MOVL (R1)+,(R0)+ ;INSERT DATAPATH REGISTER
80 81 D0 07AD 1444 MOVL (R1)+,(R0)+ ;INSERT FINAL MAP REGISTER
80 61 D0 07B0 1445 MOVL (R1),(R0)+ ;INSERT PREVIOUS MAP REGISTER
05 07B3 1446 RSB ;
```

```
07B4 1448 .SBTTL RK06/RK07 DISK DRIVE INITIALIZATION
07B4 1449 :
07B4 1450 : DM_RK0X_INIT - RK06/RK07 DISK DRIVE INITIALIZATION
07B4 1451 :
07B4 1452 : THIS ROUTINE IS CALLED AT SYSTEM INITIALIZATION AND AT POWER RECOVERY TO SET
07B4 1453 : DRIVE PARAMETERS AND TO WAIT FOR ONLINE DRIVES TO SPIN UP.
07B4 1454 :
07B4 1455 : INPUTS:
07B4 1456 :
07B4 1457 : R4 = ADDRESS OF CONTROL STATUS REGISTER 1.
07B4 1458 : R5 = DEVICE UNIT UCB ADDRESS.
07B4 1459 :
07B4 1460 : OUTPUTS:
07B4 1461 :
07B4 1462 : UNIT PARAMETERS ARE ESTABLISHED AND THE DRIVE IS SPUN UP IF IT WAS ONLINE.
07B4 1463 :
07B4 1464 :
07B4 1465 DM_RK0X_INIT: ;RK06/RK07 UNIT INITIALIZATION
07B4 1466 MOVW #RK_CS1_M_CERR,RK_CS1(R4) ;CLEAR CONTROLLER ERRORS
07B4 1467 MOVW UCB$W_UNIT(R5),RK_CS2(R4) ;SET UNIT NUMBER
07B4 1468 MOVW #F_DRVCLR!1,RK_CS1(R4) ;CLEAR DRIVE AND OBTAIN STATUS
07B4 1469 BSBW DM_WAIT ;WAIT FOR FUNCTION TO COMPLETE
07B4 1470 BSBW DM_DTYPE ;CLASSIFY DRIVE TYPE
07B4 1471 MOVZWL UCB$W_STS(R5),R3 ;SAVE CURRENT UNIT STATUS
07B4 1472 BICW #UCB$M_ONLINE!UCB$M_VALID,UCB$W_STS(R5) ;SET UNIT OFFLINE/INVALID
07B4 1473 BITW #RK_CS2_M_NED,RK_CS2(R4) ;NONEXISTENT DISK?
07B4 1474 BNEQ 50$ ;IF NEQ YES
07B4 1475 BISW #UCB$M_ONLINE,UCB$W_STS(R5) ;SET UNIT ONLINE
07B4 1476 BBC #UCB$V_VALID,R3,40$ ;IF CLR, VOLUME SOFTWARE INVALID
07B4 1477 10$: MOVW #RK_CST_M_CERR,RK_CS1(R4) ;CLEAR CONTROLLER ERRORS
07B4 1478 MOVW UCB$W_UNIT(R5),RK_CS2(R4) ;SET UNIT NUMBER
07B4 1479 BISW3 UCB$W_DM_DTYP(R5),#F_DRVCLR!1,RK_CS1(R4) ;CLEAR DRIVE
07B4 1480 BSBW DM_WAIT ;WAIT FOR FUNCTION TO COMPLETE
07B4 1481 BITW #RK_CS1_M_CERR,RK_CS1(R4) ;ANY CONTROLLER ERRORS?
07B4 1482 BNEQ 20$ ;IF NEQ YES
07B4 1483 BITW #RK_DS_M_DRDY,RK_DS(R4) ;DRIVE READY?
07B4 1484 BNEQ 30$ ;IF NEQ YES
07B4 1485 20$: JSB G^EXESPWRTIMCHK ;CHECK FOR MAXIMUM TIME EXCEEDED
07B4 1486 BLBS R0,10$ ;IF LBS MORE TIME TO GO
07B4 1487 BRB 40$ ;
07B4 1488 30$: BISW3 UCB$W_DM_DTYP(R5),#F_PACKACK!1,RK_CS1(R4) ;ACKNOWLEDGE PACK
07B4 1489 BSBW DM_WAIT ;WAIT FOR FUNCTION TO COMPLETE
07B4 1490 BITW #RK_CS1_M_CERR,RK_CS1(R4) ;ANY CONTROLLER ERRORS?
07B4 1491 BNEQ 40$ ;IF NEQ YES
07B4 1492 BISW #UCB$M_VALID,UCB$W_STS(R5) ;SET VOLUME SOFTWARE VALID
07B4 1493 40$: MOVW #RK_CST_M_CERR,RK_CS1(R4) ;CLEAR CONTROLLER ERRORS
07B4 1494 MOVW UCB$W_UNIT(R5),RK_CS2(R4) ;SET UNIT NUMBER
07B4 1495 BISW3 UCB$W_DM_DTYP(R5),#F_DRVCLR!1,RK_CS1(R4) ;CLEAR DRIVE
07B4 1496 BSBW DM_WAIT ;WAIT FOR FUNCTION TO COMPLETE
07B4 1497 50$: MOVW #RK_CS1_M_CERR,RK_CS1(R4) ;CLEAR CONTROLLER ERRORS
07B4 1498 MOVZBW #RK_CS1_M_IE,RK_CS1(R4) ;ENABLE DEVICE INTERRUPTS
07B4 1499 RSB ;
```

64	8000	8F	B0
08 A4	54 A5		B0
	64 05		B0
	0098		30
	FF7C		30
53	64 A5		3C
64 A5	0810 8F	AA	07CB
08 A4	1000 8F	B3	07D1
	5E	12	07D7
64 A5	10	A8	07D9
43 53	0B	E1	07DD
64 8000	8F	B0	07E1
08 A4	54 A5	B0	07E6
64 05	00CC C5	A9	07EB
	0068	30	07F1
64 8000	8F	B3	07F4
	08	12	07F9
0A A4	0080 8F	B3	07FB
	0B	12	0801
	00000000 GF	16	0803
	D5 50	E8	0809
	16	11	080C
64 03	00CC C5	A9	080E
	0045	30	0814
64 8000	8F	B3	0817
	06	12	081C
64 A5	0800 8F	A8	081E
64 8000	8F	B0	0824
08 A4	54 A5	B0	0829
64 05	00CC C5	A9	082E
	0025	30	0834
64 8000	8F	B0	0837
64 40	8F	9B	083C
		05	0840

```
0841 1501 .SBTTL RK611-RK06/RK07 UNSOLICITED INTERRUPT ROUTINE
0841 1502 :
0841 1503 : DM_UNSolNT - RK611-RK06/RK07 UNSOLICITED INTERRUPT ROUTINE
0841 1504 :
0841 1505 : THIS ROUTINE IS CALLED WHEN AN UNSOLICITED ATTENTION CONDITION IS DETECTED FOR
0841 1506 : AN RK06 OR RK07 DRIVE.
0841 1507 :
0841 1508 : INPUTS:
0841 1509 :
0841 1510 : R4 = ADDRESS OF CONTROL STATUS REGISTER 1.
0841 1511 : R5 = DEVICE UNIT UCB ADDRESS.
0841 1512 :
0841 1513 : OUTPUTS:
0841 1514 :
0841 1515 : IF VOLUME VALID IS CLEAR, THEN SOFTWARE VOLUME VALID IS CLEARED. THE
0841 1516 : UNIT STATUS IS CHANGED TO ONLINE AND THE DRIVE TYPE AND PARAMETERS ARE
0841 1517 : CLASSIFIED.
0841 1518 :
0841 1519 :
0841 1520 DM_UNSolNT: ;RK611-RK06/RK07 UNSOLICITED INTERRUPT
0841 1521 B1SW #UCBSM_ONLINE,UCBSW_STS(R5) ;SET UNIT ONLINE
0841 1522 BSBW DM_DTYPE ;CLASSIFY DRIVE TYPE
0841 1523 BBC #UCBSV_VALID,UCBSW_STS(R5),10$ ;IF CLR, VOLUME SOFTWARE INVALID
0841 1524 BITW #RK_DS_M_DRDY,RK_DS(R4) ; DRIVE READY BIT SET?
0841 1525 BNEQ 10$ ;IF NEQ YES
0841 1526 BICW #UCBSM_VALID,UCBSW_STS(R5) ;CLEAR SOFTWARE VOLUME VALID
0841 1527 10$: RSB ;
```

64	A5	10	A8	0841	1521	B1SW	#UCBSM_ONLINE,UCBSW_STS(R5) ;SET UNIT ONLINE
		FEFB	30	0845	1522	BSBW	DM_DTYPE ;CLASSIFY DRIVE TYPE
0E	64	A5	E1	0848	1523	BBC	#UCBSV_VALID,UCBSW_STS(R5),10\$ ;IF CLR, VOLUME SOFTWARE INVALID
0A	A4	0080	B3	084D	1524	BITW	#RK_DS_M_DRDY,RK_DS(R4) ; DRIVE READY BIT SET?
		06	12	0853	1525	BNEQ	10\$ ;IF NEQ YES
64	A5	0800	AA	0855	1526	BICW	#UCBSM_VALID,UCBSW_STS(R5) ;CLEAR SOFTWARE VOLUME VALID
		8F	05	085B	1527	RSB	;

```
085C 1529 .SBTTL WAIT FOR CONTROLLER READY
085C 1530 :
085C 1531 : DM_WAIT - WAIT FOR CONTROLLER READY
085C 1532 :
085C 1533 : THIS SUBROUTINE IS CALLED TO POLL THE RK611 CONTROLLER FOR READY. A MAX-
085C 1534 : IMUM OF APPROXIMATELY 20 US ELAPSES BEFORE CONTROL IS RETURNED TO THE
085C 1535 : CALLER.
085C 1536 :
085C 1537 :
085C 1538 DM_WAIT: ;WAIT FOR CONTROLLER READY
7E 50 7D 085C 1539 MOVQ R0,-(SP) ;SAVE R0, R1
085F 1540 DSBINT ;DISABLE INTERRUPTS
0865 1541 TIMEWAIT #2,WRK_CS1_M_RDY,RK_CS1(R4),W
088A 1542 ENBINT ;ENABLE INTERRUPTS
50 8E 7D 088D 1543 MOVQ (SP)+,R0 ;RESTORE R0, R1
05 0890 1544 RSB ;
```

```
0891 1546 .SBTTL RK611 DISK CONTROLLER INTERRUPT DISPATCHER
0891 1547 :+
0891 1548 : DMSINT - RK611 DISK CONTROLLER INTERRUPT DISPATCHER
0891 1549 :
0891 1550 : THIS ROUTINE IS ENTERED VIA A JSB INSTRUCTION WHEN AN INTERRUPT OCCURS
0891 1551 : ON AN RK611 DISK CONTROLLER. THE STATE OF THE STACK ON ENTRY IS:
0891 1552 :
0891 1553 : 00(SP) = ADDRESS OF IDB ADDRESS.
0891 1554 : 04(SP) = SAVED R2.
0891 1555 : 08(SP) = SAVED R3.
0891 1556 : 12(SP) = SAVED R4.
0891 1557 : 16(SP) = SAVED R5.
0891 1558 : 20(SP) = INTERRUPT PC.
0891 1559 : 24(SP) = INTERRUPT PSL.
0891 1560 :
0891 1561 : INTERRUPT DISPATCHING OCCURS AS FOLLOWS:
0891 1562 :
0891 1563 : IF THE INTERRUPTING CONTROLLER IS CURRENTLY OWNED AND THE OWNER
0891 1564 : UNIT IS EXPECTING AN INTERRUPT, THEN THAT UNIT IS DISPATCHED FIRST.
0891 1565 : ALL OTHER UNITS ARE DISPATCHED BY READING THE ATTENTION SUMMARY
0891 1566 : REGISTER AND SCANNING FOR UNITS THAT HAVE ATTENTION SET. AS EACH
0891 1567 : UNIT IS FOUND, A TEST IS MADE TO DETERMINE IF AN INTERRUPT IS
0891 1568 : EXPECTED ON THE UNIT. IF YES, THEN THE DRIVER IS CALLED AT ITS
0891 1569 : INTERRUPT RETURN ADDRESS. ELSE THE DRIVER IS CALLED AT ITS UNSOL-
0891 1570 : ICITED INTERRUPT ADDRESS. AS EACH CALL TO THE DRIVER RETURNS, THE
0891 1571 : ATTENTION SUMMARY REGISTER IS REREAD AND AN ATTEMPT IS MADE TO FIND
0891 1572 : ANOTHER UNIT TO DISPATCH. WHEN NO UNITS REQUESTING ATTENTION REMAIN,
0891 1573 : THE INTERRUPT IS DISMISSED.
0891 1574 : -
0891 1575 :
0891 1576 DMSINT:: :RK611 DISK CONTROLLER INTERRUPT DISPATCHER
53 00 BE D0 0891 1577 MOVL @ (SP),R3 :GET ADDRESS OF IDB
54 63 D0 0895 1578 MOVL IDBSL_CSR(R3),R4 :GET ADDRESS OF CONTROL STATUS REGISTER 1
55 04 A3 D0 0898 1579 MOVL IDBSL_OWNER(R3),R5 :GET OWNER UNIT UCB ADDRESS
05 13 089C 1580 BEQL 10$ :IF EQL NO OWNER
3D 64 A5 01 E4 089E 1581 BBSC #UCBSW_INT,UCBSW_STS(R5),30$ :IF SET, INTERRUPT EXPECTED
64 8000 8F B0 08A3 1582 10$: MOVW #RK_CST_M_CERR,RK_CS1(R4) :CLEAR CONTROLLER
52 52 0E A4 3C 08A8 1583 MOVZWL RK_AS(R4),R2 :READ ATTENTION SUMMARY REGISTER
08 08 EA 08AC 1584 FFS #8,#8,R2,R2 :FIND FIRST UNIT REQUESTING ATTENTION
11 12 08B1 1585 BNEQ 20$ :IF NEQ UNIT FOUND
64 40 8F 9B 08B3 1586 MOVZBW #RK_CS1_M_IE,RK_CS1(R4) :ENABLE DEVICE INTERRUPTS
5E 04 C0 08B7 1587 ADDL #4,SP :CLEAN STACK
50 8E 7D 08BA 1588 MOVQ (SP)+,R0 :RESTORE REGISTERS
52 8E 7D 08BD 1589 MOVQ (SP)+,R2
54 8E 7D 08C0 1590 MOVQ (SP)+,R4
02 08C3 1591 REI
08 52 08 C2 08C4 1592 20$: SUBL #8,R2 :CALCULATE UNIT NUMBER
08 A4 52 B0 08C7 1593 MOVW R2,RK_CS2(R4) :SET UNIT NUMBER
55 18 A3 42 D0 08CB 1594 MOVL IDBSL_UCBLST(R3)[R2],R5 :GET ADDRESS OF UCB
79 13 08D0 1595 BEQL 80$ :IF EQL NO CORRESPONDING UNIT
64 01 00CC C5 A9 08D2 1596 BISW3 UCBSW_DM_DTYP(R5),#F_NOP!1,RK_CS1(R4) :SELECT DRIVE AND GET STATUS
FF 81 30 08D8 1597 BSBW DM_WAIT :WAIT FOR CONTROLLER READY
6D 64 A5 01 E5 08DB 1598 BBCC #UCBSW_INT,UCBSW_STS(R5),90$ :IF CLR, INTERRUPT NOT EXPECTED
0093 C5 0E 91 08E0 1599 30$: CMPB #CDF_READHEAD,UCBSB_CEX(R5) :READ HEADER FUNCTION?
12 12 08E5 1600 BNEQ 40$ :IF NEQ NO
00F4 C5 14 A4 B0 08E7 1601 MOVW RK_DB(R4),UCBSW_DM_DB(R5) :SAVE SECTOR HEADER INFORMATION
00F6 C5 14 A4 B0 08ED 1602 MOVW RK_DB(R4),UCBSW_DM_DB+2(R5) :
```

```
00F8 C5 14 A4 B0 08F3 1603 MOVW RK_DB(R4),UCBSW_DM_DB+4(R5);
53 52 64 9E 08F9 1604 40$: MOVAB RK_CS1(R4),R2;GET ADDRESS OF CONTROL STATUS REGISTER 1
00CE C5 9E 08FC 1605 MOVAB UCBSW_DM_CS1(R5),R3;GET ADDRESS OF REGISTER SAVE AREA
83 82 B0 0901 1606 MOVW (R2)+,(R3)+;SAVE CONTROL STATUS REGISTER 1
7D 19 0904 1607 BLSS 120$;IF LSS ERROR ENCOUNTERED
78 68 A5 01 E0 0906 1608 BBS #UCBSV_DIAGBUF,UCBSW_DEVSTS(R5),120$;IF SET, DIAGNOSTIC BUFFER
08 91 0908 1609 CMPB #CDF_PACKACK,-;PACK ACKNOWLEDGE FUNCTION?
0093 C5 090D 1610 UCBSB_CEX(R5)
0F 12 0910 1611 BNEQ 50$;BRANCH IF NOT.
0A A4 0080 8F B3 0912 1612 BITW #RK_DS_M_DRDY,RK_DS(R4);DRIVE READY BIT SET?
69 13 0918 1613 BEQL 120$;ERROR IF NOT.
0080 8F AB 091A 1614 BISW #RK_DS_M_DRDY,-;SAVE READY BIT IN UCB.
00D8 C5 091E 1615 UCBSW_DM_DS(R5)
83 82 B0 0921 1616 50$: MOVW (R2)+,(R3)+;SAVE WORD COUNT REGISTER
63 62 B0 0924 1617 MOVW (R2),(R3);SAVE BUFFER ADDRESS REGISTER
53 10 A5 7D 0927 1618 60$: MOVQ UCBSL_FR3(R5),R3;RESTORE DRIVER CONTEXT
0C B5 16 0928 1619 JSB @UCBSL_FPC(R5);CALL DRIVER AT INTERRUPT RETURN ADDRESS
53 00 BE D0 092E 1620 MOVL @ (SP),R3;GET ADDRESS OF IDB
54 63 D0 0932 1621 MOVL IDBSL_CSR(R3),R4;GET ADDRESS OF CONTROL STATUS REGISTER 1
64 8000 8F B0 0935 1622 70$: MOVW #RK_CS1_M_CERR,RK_CS1(R4);CLEAR CONTROLLER
08 A4 54 A5 B0 093A 1623 MOVW UCBSW_UNIT(R5),RK_CS2(R4);SET UNIT NUMBER
64 05 00CC C5 A9 093F 1624 BISW3 UCBSW_DM_DTYP(R5),#F_DRVCLR!1,RK_CS1(R4);CLEAR DRIVE ERRORS
FF14 30 0945 1625 BSBW DM_WAIT;WAIT FOR CONTROLLER READY
FF58 31 0948 1626 BRW 10$;
18 11 094B 1627 80$: BRB 100$;
FEF1 30 094D 1628 90$: BSBW DM_UNSLNT;CALL UNSOLICITED INTERRUPT ROUTINE
53 00 BE D0 0950 1629 MOVL @ (SP),R3;GET ADDRESS OF IDB
54 63 D0 0954 1630 MOVL IDBSL_CSR(R3),R4;GET ADDRESS OF CONTROL STATUS REGISTER 1
64 8000 8F B3 0957 1631 BITW #RK_CS1_M_CERR,RK_CS1(R4);ANY ERROR CONDITION PRESENT?
D7 13 095C 1632 BEQL 70$;IF EQL NO
08 A4 20 B0 095E 1633 MOVW #RK_CS2_M_SCLR,RK_CS2(R4);CLEAR ENTIRE RK611 SUBSYSTEM
FF3E 31 0962 1634 BRW 10$;
64 0405 8F B0 0965 1635 100$: MOVW #RK_CS1_M_CDT!F_DRVCLR!1,RK_CS1(R4);CLEAR RK07 DRIVE
FEEF 30 096A 1636 BSBW DM_WAIT;WAIT FOR FUNCTION TO COMPLETE
64 64 0F B5 096D 1637 TSTW RK_CS1(R4);SUCCESSFUL COMPLETION?
8000 8F B0 0971 1639 MOVW #RK_CS1_M_CERR,RK_CS1(R4);IF GEQ YES
08 A4 52 B0 0976 1640 MOVW R2,RK_CS2(R4);CLEAR CONTROLLER
64 05 B0 097A 1641 MOVW #F_DRVCLR!1,RK_CS1(R4);SET UNIT NUMBER
FEDC 30 097D 1642 BSBW DM_WAIT;CLEAR RK06 DRIVE
FF20 31 0980 1643 110$: BRW 10$;WAIT FOR FUNCTION TO COMPLETE
83 82 B0 0983 1644 120$: MOVW (R2)+,(R3)+;SAVE WORD COUNT REGISTER
83 82 B0 0986 1645 MOVW (R2)+,(R3)+;SAVE BUFFER ADDRESS REGISTER
83 82 B0 0989 1646 MOVW (R2)+,(R3)+;SAVE DESIRED SECTOR/TRACK ADDRESS REGISTER
83 82 B0 098C 1647 MOVW (R2)+,(R3)+;SAVE CONTROL STATUS REGISTER 2
83 82 B0 098F 1648 MOVW (R2)+,(R3)+;SAVE DRIVE STATUS REGISTER
83 82 B0 0992 1649 MOVW (R2)+,(R3)+;SAVE ERROR REGISTER
83 82 B0 0995 1650 MOVW (R2)+,(R3)+;SAVE ATTENTION SUMMARY/OFFSET REGISTER
83 82 B0 0998 1651 MOVW (R2)+,(R3)+;SAVE DESIRED CYLINDER ADDRESS REGISTER
52 04 C0 099B 1652 ADDL #4,R2;POINT TO MAINTENANCE REGISTER 1
83 82 B0 099E 1653 MOVW (R2)+,(R3)+;SAVE MAINTENANCE REGISTER 1
00C4 C5 82 B0 09A1 1654 MOVW (R2)+,UCBSW_EC1(R5);SAVE ECC POSITION REGISTER
00C6 C5 82 B0 09A6 1655 MOVW (R2)+,UCBSW_EC2(R5);SAVE ECC PATTERN REGISTER
83 82 B0 09AB 1656 MOVW (R2)+,(R3)+;SAVE MAINTENANCE REGISTER 2
63 62 B0 09AE 1657 MOVW (R2),(R3);SAVE MAINTENANCE REGISTER 3
04 00D6 C5 09 E1 09B1 1658 BBC #RK_CS2_V_MDS,UCBSW_DM_CS2(R5),130$;IF CLR, NO MULTI-DRIVE SELECT
08 A4 20 B0 09B7 1659 MOVW #RK_CS2_M_SCLR,RK_CS2(R4);CLEAR ENTIRE SUBSYSTEM
```

DMDRIVER  
V04-000

- RK611-RK06/RK07 DISK DRIVER  
RK611 DISK CONTROLLER INTERRUPT DISPATCH

H 11

15-SEP-1984 23:47:21  
5-SEP-1984 00:12:35

VAX/VMS Macro V04-00  
[DRIVER.SRC]DMDRIVER.MAR;1

Page 34  
(1)

FF69 31 09BB 1660 130\$: BRW 60\$

;

```

09BE 1662 .SBTTL RK611 DISK CONTROLLER INITIALIZATION
09BE 1663 :+
09BE 1664 : DM_RK611_INIT - RK611 DISK CONTROLLER INITIALIZATION
09BE 1665 :
09BE 1666 : THIS ROUTINE IS CALLED VIA A JSB INSTRUCTION AT SYSTEM STARTUP AND AFTER
09BE 1667 : A POWER RECOVERY RESTART TO ALLOW INITIALIZATION OF RK611 DISK CONTROLLERS.
09BE 1668 :
09BE 1669 : INPUTS:
09BE 1670 :
09BE 1671 : R0 = SCRATCH.
09BE 1672 : R1 = SCRATCH.
09BE 1673 : R2 = SCRATCH.
09BE 1674 : R3 = SCRATCH.
09BE 1675 : R4 = ADDRESS OF CONTROL STATUS REGISTER 1.
09BE 1676 : R5 = ADDRESS OF CONTROLLER IDB.
09BE 1677 :
09BE 1678 : ALL INTERRUPTS ARE LOCKED OUT.
09BE 1679 :
09BE 1680 : OUTPUTS:
09BE 1681 :
09BE 1682 : THE RK611 CONTROLLER IS INITIALIZED AND INTERRUPTS ARE ENABLED.
09BE 1683 : -
09BE 1684 :
09BE 1685 DM_RK611_INIT: ;RK611 DISK CONTROLLER INITIALIZATION
08 A4 20 B0 09BE 1686 MOVW #RK_CS2_M_SCLR,RK_CS2(R4) ;CLEAR CONTROLLER AND ALL DRIVES
64 40 8F 9B 09C2 1687 MOVZBW #RK_CS1_M_IE,RK_CS1(R4) ;ENABLE DEVICE INTERRUPTS
05 09C6 1688 RSB ;

```

```
09C7 1690 .SBTTL RK611 Autoconfigure Unit Delivery Routine
09C7 1691 :+
09C7 1692 : DMSDELIVER - RK611 Autoconfigure Unit Delivery Routine
09C7 1693 :
09C7 1694 : This routine is called by the SYSGEN AUTOCONFIGURE command to determine
09C7 1695 : which RK611 unit numbers to configure. It is called once for each possible
09C7 1696 : unit, 0 through 7.
09C7 1697 :
09C7 1698 : INPUTS:
09C7 1699 : R3 Address of controller IDB, or zero if none exists
09C7 1700 : R4 Address of CSR
09C7 1701 : R5 Unit number which this routine must decide whether or not to
09C7 1702 : configure
09C7 1703 : R6 Base address of UNIBUS adapter I/O space
09C7 1704 : R7 Address of AUTOCONFIGURE ACF
09C7 1705 : R8 Address of UNIBUS ADP
09C7 1706 : IPL = 31
09C7 1707 :
09C7 1708 : OUTPUTS:
09C7 1709 : R0 TRUE ==> configure unit indicated in R5
09C7 1710 : FALSE ==> do not configure unit indicated in R5
09C7 1711 :
09C7 1712 : Interference with "normal" data transfers is a major concern for this unit
09C7 1713 : delivery routine. Since it is called without the controls of the QIO
09C7 1714 : mechanism, the state of data transfers when it is entered is unpredictable.
09C7 1715 : Experience has shown that conditions are so unpredictable that the only
09C7 1716 : option open to this routine is forcing all current activity to be retried.
09C7 1717 : To this end, all UCBs listed in the IDB passed to this routine, if any, are
09C7 1718 : made to appear as if a power failure has occurred. Having done this, the
09C7 1719 : retrying of currently active operations is relative assured. We do not
09C7 1720 : simulate a power failure to the extent of calling the controller and unit
09C7 1721 : initialization routines; after all, this routine determines the state of the
09C7 1722 : controller and its units quite completely.
09C7 1723 :
09C7 1724 :-
09C7 1725 :
09C7 1726 DMSDELIVER:
55 D5 09C7 1727 TSTL R5 ; Is this the first call for this
02 12 09C9 1728 BNEQ 10$ ; for this controller? If so, get
07 10 09CB 1729 BSBB GET_UNITS ; complete units present information.
50 24 A7 01 55 EF 09CD 1730 10$: EXTZV R5, #1, - ; For each unit, get presense data from
09D3 1731 ACF$$_DLVR_SCRH(R7), R0 ; information prepared by GET_UNITS.
05 09D3 1732 RSB ; Then, return to AUTOCONFIGURE.
09D4 1733 :
09D4 1734 :+
09D4 1735 : NB: the use of the one-time get-units-information routine GET_UNITS reduces
09D4 1736 : to one the number of times we must fool with the controller and thus our
09D4 1737 : potential for munging "normal" operations. It also insures that all
09D4 1738 : controller munging occurs at the same time that a power failure is
09D4 1739 : simulated.
09D4 1740 :-
09D4 1741 :
09D4 1742 GET_UNITS:
09D4 1743 DSBINT ; Insure no interruptions.
09DA 1744 :
09DA 1745 :+
09DA 1746 : SIMULATE A POWER FAILURE ON ALL KNOWN UCBs
```

```

53 D5 09DA 1747 :-
23 13 09DA 1748 TSTL R3 ; Is there an IDB? If not, the RK611
50 OC A3 3C 09DC 1749 BEQL 500$ ; must be inactive.
51 14 A340 D0 09DE 1750 MOVZWL IDBSW_UNITS(R3), R0 ; Get count of UCBs to test.
64 A1 15 13 09E2 1751 10$: MOVL IDBSL_UCBLST-4(R3)(R0), R1 ; Get a UCB address.
64 A1 20 A8 09E7 1752 BEQL 19$ ; Branch if not really a UCB address.
64 A1 03 B3 09E9 1753 BISW #UCBSM_POWER, UCBSW_STS(R1) ; Pretend power failure.
64 A1 03 B3 09ED 1754 BITW #<UCBSM_INT!UCBSM_TIM>, - ; Test for timeout in progress.
0B 13 09F1 1755 UCBSW_STS(R1)
64 A1 02 AA 09F3 1756 BEQL 19$ ; Branch if no timeout in progress.
64 A1 01 A8 09F7 1757 BICW #UCBSM_INT, UCBSW_STS(R1) ; Clear interrupt expected.
6C A1 D4 09FB 1758 BISW #UCBSM_TIM, UCBSW_STS(R1) ; Indicate that a timeout is expected
E1 50 F5 09FE 1759 CLRL UCBSL_DUETIME(R1) ; immediately.
19$: SOBGTR R0, 10$ ; Loop through all UCBs.

0A01 1761
0A01 1762 :+
0A01 1763 : DISCOVERING WHICH UNITS ARE PRESENT
0A01 1764 :-
24 A7 D4 0A01 1765 500$: CLRL ACFSL_DLVR_SCRH(R7) ; Clear all units present bits.
50 D4 0A04 1766 CLRL R0 ; Initialize unit number.
FE53 30 0A06 1767 600$: BSBW DM_WAIT ; Wait for controller ready.
64 8000 8F B0 0A09 1768 MOVW #RR_CS1_M_CERR, RK_CS1(R4) ; Clear controller.
08 A4 50 B0 0A0E 1769 RO, RK_CS2(R4) ; Set unit number.
64 01 B0 0A12 1770 MOVW #1, RK_CS1(R4) ; Select drive and get status.
FE44 30 0A15 1771 BSBW DM_WAIT ; Wait for controller ready.
08 A4 1000 8F B3 0A18 1772 BITW #RR_CS2_M_NED, RK_CS2(R4) ; Nonexistent drive?
26 12 0A1E 1773 BNEQ 690$ ; If nonexistent, no more to do here.
00 24 A7 50 E2 0A20 1774 BBSS R0, ACFSL_DLVR_SCRH(R7), 610$ ; Set device present bit.
64 8000 8F B0 0A25 1775 610$: MOVW #RR_CS1_M_CERR, RK_CS1(R4) ; Clear controller.
08 A4 50 B0 0A2A 1776 MOVW RO, RK_CS2(R4) ; Set unit number.
64 0405 8F B0 0A2E 1777 MOVW #X405, RK_CS1(R4) ; Clear drive as a RK07.
FE26 30 0A33 1778 BSBW DM_WAIT ; Wait for function to complete.
64 0C B5 0A36 1779 TSTW RK_CS1(R4) ; Controller errors? Errors mean its a
64 8000 8F B0 0A38 1780 BGEQ 690$ ; RK06 and must be cleared differently.
08 A4 50 B0 0A3A 1781 MOVW #RR_CS1_M_CERR, RK_CS1(R4) ; Clear controller.
64 05 B0 0A3F 1782 MOVW RO, RK_CS2(R4) ; Set unit number.
BC 50 07 B0 0A43 1783 MOVW #5, RK_CS1(R4) ; Clear drive as a RK06.
FE0F 30 0A46 1784 690$: AOBLEQ #7, R0, 600$ ; Loop over all possible drives.
64 8000 8F B0 0A4A 1785 BSBW DM_WAIT ; Wait for last operation to complete.
05 0A52 1786 MOVW #RR_CS1_M_CERR, RK_CS1(R4) ; Clear controller.
0A55 1787 ENBINT ; Restore previous interrupt state.
0A56 1788 RSB ; Return to main unit-deliver routine.
0A56 1789
0A56 1790 DM_END: ; ADDRESS OF LAST LOCATION IN DRIVER
0A56 1791
0A56 1792 .END
```

DMDRIVER  
Symbol table

- RK611-RK06/RK07 DISK DRIVER

L 11

15-SEP-1984 23:47:21 VAX/VMS Macro V04-00  
5-SEP-1984 00:12:35 [DRIVER.SRC]DMDRIVER.MAR;1

Page 38  
(1)

```

$$$ = 00000020 R 02
$$OP = 00000002
ACFSL_DLVR_SCRH = 00000024
ACPSACCESS ***** X 03
ACPSDEACCESS ***** X 03
ACPSMODIFY ***** X 03
ACPSMOUNT ***** X 03
ACPSREADBLK ***** X 03
ACPSWRITEBLK ***** X 03
APPLY_ECC = 000002A7 R 03
ATS_UBA = 00000001
AVAILABLE = 000001C8 R 03
CDF_AVAILABLE = 0000000F
CDF_DRVCLR = 00000004
CDF_NOP = 00000000
CDF_OFFSET = 00000006
CDF_PACKACK = 00000008
CDF_READDATA = 0000000C
CDF_READHEAD = 0000000E
CDF_RECAL = 00000003
CDF_RELEASE = 00000005
CDF_RETCENTER = 00000007
CDF_SEEK = 00000002
CDF_STARTSPNDL = 00000009
CDF_UNLOAD = 00000001
CDF_WRITECHECK = 0000000A
CDF_WRITEDATA = 0000000B
CDF_WRITEHEAD = 0000000D
CHECKRETRY = 00000237 R 03
CHECKXT = 00000241 R 03
CRBSL_INTD = 00000024
DATAHECK = 00000209 R 03
DCS_DISK = 00000001
DDBSK_CART = 00000002
DDBSL_ACPD = 00000010
DDBSL_DDT = 0000000C
DEFER_ECC = 000002D2 R 03
DEVSM_AVL = 00040000
DEVSM_DIR = 00000008
DEVSM_ELG = 00400000
DEVSM_FOD = 00004000
DEVSM_IDV = 04000000
DEVSM_NNM = 00000200
DEVSM_ODV = 08000000
DEVSM_RND = 10000000
DEVSM_SHR = 00010000
DMSDDT = 00000000 RG 03
DMSDELIVER = 000009C7 R 03
DMSINT = 00000891 RG 03
DM_BYTECNT = 00000100 R 03
DM_DTYPE = 00000743 R 03
DM_END = 00000A56 R 03
DM_FUNCABLE = 00000060 R 03
DM_IND_M_OF = 00000001
DM_IND_V_OF = 00000000
DM_M_DCK = 00000002
DM_M_ECC_DEFER = 00000004

```

```

DM_M_ECI
DM_REGDUMP
DM_RK0X_INIT
DM_RK61T_INIT
DM_STARTTO
DM_UNSLNT
DM_V_DCK
DM_V_ECC_DEFER
DM_V_ECI
DM_WAIT
DPTSC_LENGTH
DPTSC_VERSION
DPTSCINITAB
DPTSM_SVP
DPTSCREINITAB
DPTSTAB
DRVCLR
DTS_RK06
DTS_RK07
DYNSC_CRB
DYNSC_DDB
DYNSC_DPT
DYNSC_UCB
ECC
EMBSL_DV_REGSAR
ENBXIT
ERLSDVICERR
ERLSDVICIMO
EXESABORTIO
EXESGL_TENUSEC
EXESGL_UBDELAY
EXESIOFORK
EXESLCLDSKVALID
EXESONEPARM
EXESPWRTIMCHK
EXESSENSEMODE
EXESSETCHAR
EXESZEROPARM
EXEC_FUNCTION
FATALERR
FDISPATCH
FEXH
FEXL
FTAB
FUNCTAB_LEN
FUNCXT
F_AVAILABLE
F_DRVCLR
F_NOP
F_OFFSET
F_PACKACK
F_READDATA
F_READHEAD
F_RECAL
F_RELEASE
F_RETCENTER
F_SEEK

```

```

= 00000001
00000786 R 03
000007B4 R 03
000009BE R 03
00000111 R 03
00000841 R 03
= 00000001
= 00000002
= 00000000
0000085C R 03
= 00000038
= 00000004
00000038 R 02
= 00000002
00000072 R 02
00000000 R 02
000001DE R 03
= 00000001
= 00000002
= 00000005
= 00000006
= 0000001E
= 00000010
0000026A R 03
= 0000004E
000004DC R 03
***** X 03
***** X 03
***** X 03
***** X 03
***** X 03
***** X 03
***** X 03
***** X 03
***** X 03
***** X 03
***** X 03
000001DE R 03
0000034B R 03
0000017E R 03
00000406 R 03
0000040F R 03
00000038 R 03
= 000000A0
000003C1 R 03
= 00000000
= 00000004
= 00000000
= 0000000C
= 00000002
= 00000010
= 00000014
= 0000000A
= 00000000
= 0000000C
= 0000000E

```

DMDRIVER  
Symbol table

- RK611-RK06/RK07 DISK DRIVER

M 11

15-SEP-1984 23:47:21 VAX/VMS Macro V04-00  
5-SEP-1984 00:12:35 [DRIVER.SRC]DMDRIVER.MAR;1

Page 39  
(1)

F_STARTSPNDL	= 00000008		
F_UNLOAD	= 00000006		
F_WRITECHECK	= 00000018		
F_WRITEDATA	= 00000012		
F_WRITEHEAD	= 00000016		
GET_UNITS	= 000009D4	R	03
IDBSL_CSR	= 00000000		
IDBSL_OWNER	= 00000004		
IDBSL_UCBLST	= 00000018		
IDBSW_UNITS	= 0000000C		
IMMED	= 00000444	R	03
IOSM_DATACHECK	= 00004000		
IOSV_DATACHECK	= 0000000E		
IOSV_INHRETRY	= 0000000F		
IOSV_INHSEEK	= 0000000C		
IOS_ACCESS	= 00000032		
IOS_ACPCONTROL	= 00000038		
IOS_AVAILABLE	= 00000011		
IOS_CREATE	= 00000033		
IOS_DEACCESS	= 00000034		
IOS_DELETE	= 00000035		
IOS_DRVCLR	= 00000004		
IOS_MODIFY	= 00000036		
IOS_MOUNT	= 00000039		
IOS_NOP	= 00000000		
IOS_OFFSET	= 00000006		
IOS_PACKACK	= 00000008		
IOS_READHEAD	= 0000000E		
IOS_READBLK	= 00000021		
IOS_READPBLK	= 0000000C		
IOS_READVBLK	= 00000031		
IOS_RECAL	= 00000003		
IOS_RELEASE	= 00000005		
IOS_RETCENTER	= 00000007		
IOS_SEEK	= 00000002		
IOS_SENSECHAR	= 0000001B		
IOS_SENSEMODE	= 00000027		
IOS_SETCAR	= 0000001A		
IOS_SETMODE	= 00000023		
IOS_STARTSPNDL	= 00000019		
IOS_UNLOAD	= 00000001		
IOS_VIRTUAL	= 0000003F		
IOS_WRITECHECK	= 0000000A		
IOS_WRITEHEAD	= 0000000D		
IOS_WRITEBLK	= 00000020		
IOS_WRITEPBLK	= 0000000B		
IOS_WRITEVBLK	= 00000030		
IOCSAPPLYECC	*****	X	03
IOCSDIAGBUF ILL	*****	X	03
IOCSLOADUBAMAP	*****	X	03
IOCSMNTVER	*****	X	03
IOCSMOVTOUSER	*****	X	03
IOCSPURGDATAP	*****	X	03
IOCSRELCHAN	*****	X	03
IOCSRELDATAP	*****	X	03
IOCSRELMAPREG	*****	X	03
IOCSREQCOM	*****	X	03

IOCSREQDATAP	*****	X	03
IOCSREQMAPREG	*****	X	03
IOCSREQPCHANH	*****	X	03
IOCSREQPCHANL	*****	X	03
IOCSRETURN	*****	X	03
IOCSUPDATRANSF	*****	X	03
IOCSWFIKPC	*****	X	03
IOCSWFIRLCH	*****	X	03
IRPSL_MEDIA	= 00000038		
IRPSL_SVAPTE	= 0000002C		
IRPSV_FCODE	= 00000006		
IRPSV_DIAGBUF	= 00000007		
IRPSV_FCODE	= 00000000		
IRPSV_PHYSIO	= 00000008		
IRPSW_BCNT	= 00000032		
IRPSW_FUNC	= 00000020		
IRPSW_STS	= 0000002A		
MASKH	= 00000008		
MASKL	= 04000000		
NOP	000001DE	R	03
NORMAL	0000023E	R	03
OFF	000002D7	R	03
OFFSET	000001DE	R	03
OFFSETERR	00000346	R	03
OFFSIZ	= 00000008		
OFFTAB	00000058	R	03
PACKACK	000001D8	R	03
POSIT	00000472	R	03
PRS_IPL	= 00000012		
READDATA	000001F3	R	03
READHEAD	000001E7	R	03
RECAL	000001DE	R	03
RELEASE	000001DE	R	03
RELES	00000441	R	03
RESETXFR	00000724	R	03
RETCENTER	000001DE	R	03
RETREG	00000619	R	03
RETRY	00000267	R	03
RETRYERR	0000032F	R	03
RK_AS	0000000E		
RK_BA	00000004		
RK_CS1	00000000		
RK_CS1_M_CDT	= 00000400		
RK_CS1_M_CERR	= 00008000		
RK_CS1_M_CTO	= 00000800		
RK_CS1_M_DPPE	= 00000020		
RK_CS1_M_GO	= 00000001		
RK_CS1_M_IE	= 00000040		
RK_CS1_M_RDY	= 00000080		
RK_CS1_M_SPAR	= 00002000		
RK_CS1_V_CERR	= 0000000F		
RK_CS2	00000008		
RK_CS2_M_DLT	= 00008000		
RK_CS2_M_MDS	= 00000200		
RK_CS2_M_NED	= 00001000		
RK_CS2_M_NEM	= 00000800		
RK_CS2_M_PGE	= 00000400		

DMDRIVER  
Symbol table

- RK611-RK06/RK07 DISK DRIVER

N 11

15-SEP-1984 23:47:21 VAX/VMS Macro V04-00  
5-SEP-1984 00:12:35 [DRIVER.SRC]DMDRIVER.MAR;1

Page 40  
(1)

```

RK_CS2_M_RLS      = 00000008
RK_CS2_M_SCLR     = 00000020
RK_CS2_M_UFE      = 00000100
RK_CS2_M_UPE      = 00002000
RK_CS2_M_WCE      = 00004000
RK_CS2_V_MDS      = 00000009
RK_CS2_V_NED      = 0000000C
RK_CS2_V_UPE      = 0000000D
RK_CS2_V_WCE      = 0000000E
RK_DA             = 00000006
RK_DB             = 00000014
RK_DC             = 00000010
RK_DS             = 0000000A
RK_DS_M_DDT       = 00000100
RK_DS_M_DRDY      = 00000080
RK_DS_M_DSC       = 00004000
RK_DS_M_VV        = 00000040
RK_DS_V_DRDY      = 00000007
RK_DS_V_VV        = 00000006
RK_ECT            = 00000018
RK_EC2            = 0000001A
RK_ER             = 0000000C
RK_ER_M_BSE       = 00000080
RK_ER_M_COE       = 00000200
RK_ER_M_DCK       = 00008000
RK_ER_M_DRPAR     = 00000008
RK_ER_M_DTE       = 00001000
RK_ER_M_DTYE      = 00000020
RK_ER_M_ECH       = 00000040
RK_ER_M_FMT       = 00000010
RK_ER_M_HVRC      = 00000100
RK_ER_M_IDAE      = 00000400
RK_ER_M_ILF       = 00000001
RK_ER_M_NXF       = 00000004
RK_ER_M_OPI       = 00002000
RK_ER_M_SKI       = 00000002
RK_ER_M_WLE       = 00000800
RK_ER_V_DRPAR     = 00000003
RK_ER_V_HVRC      = 00000008
RK_ER_V_UN        = 0000000E
RK_ER_V_WLE       = 0000000B
RK_MR1            = 00000016
RK_MR2            = 0000001C
RK_MR3            = 0000001E
RK_SPR            = 00000012
RK_WC             = 00000002
RLSCHN            = 00000613
SEEK              = 000001DE
SIZ...            = 00000001
SS$_CTRLERR       = 00000054
SS$_DATACHECK     = 0000005C
SS$_DRVERR        = 0000008C
SS$_FORMAT        = 0000008C
SS$_IVADDR        = 00000134
SS$_IVBUFLLEN     = 0000034C
SS$_MEDOFL        = 000001A4
SS$_NONEXDRV      = 000001C4

```

R 03  
R 03

```

SS$_NORMAL        = 00000001
SS$_PARITY        = 000001F4
SS$_TIMEOUT       = 0000022C
SS$_UNSAFE        = 0000023C
SS$_VOLINV        = 00000254
SS$_WASECC        = 00000639
SS$_WRITLCK       = 0000025C
STARTSPNDL        = 000001DE
TRANSFR           = 00000200
TRANXT            = 00000244
UCBSB_CEX         = 00000093
UCBSB_DEVCLASS    = 00000040
UCBSB_DEVTYPE     = 00000041
UCBSB_DIPL        = 0000005E
UCBSB_DM_IND      = 000000FA
UCBSB_ERTCNT      = 00000080
UCBSB_ERTMAX      = 00000081
UCBSB_FEX         = 00000092
UCBSB_FIPL        = 0000000B
UCBSB_OFFNDX      = 000000CA
UCBSB_OFFRTC      = 000000CB
UCBSB_SECTORS     = 00000044
UCBSB_TRACKS      = 00000045
UCBSK_DM_LENGTH   = 00000100
UCBSK_LCC_DISK_LENGTH = 000000CC
UCBSL_CRB         = 00000024
UCBSL_DEVCHAR     = 00000038
UCBSL_DEVCHAR2    = 0000003C
UCBSL_DM_DPR      = 000000E8
UCBSL_DM_FMPR     = 000000EC
UCBSL_DM_FRS      = 000000FB
UCBSL_DM_PMPR     = 000000F0
UCBSL_DPC         = 0000009C
UCBSL_DUETIM      = 0000006C
UCBSL_FPC         = 0000000C
UCBSL_FR3         = 00000010
UCBSL_IRP         = 00000058
UCBSL_MAXBLOCK    = 000000B0
UCBSL_MEDIA_ID    = 0000008C
UCBSL_SVAPTE      = 00000078
UCBSM_DIAGBUF     = 00000002
UCBSM_ECC         = 00000001
UCBSM_INT         = 00000002
UCBSM_ONLINE      = 00000010
UCBSM_POWER       = 00000020
UCBSM_TIM         = 00000001
UCBSM_TIMEOUT     = 00000040
UCBSM_VALID       = 00000800
UCBSV_DIAGBUF     = 00000001
UCBSV_ECC         = 00000000
UCBSV_INT         = 00000001
UCBSV_POWER       = 00000005
UCBSV_VALID       = 0000000B
UCBSW_BCNT        = 0000007E
UCBSW_BCR         = 000000C0
UCBSW_BOFF        = 0000007C
UCBSW_CYLINDERS   = 00000046

```

R 03  
R 03  
R 03

DMDRIVER  
Symbol table

- RK611-RK06/RK07 DISK DRIVER

B 12

15-SEP-1984 23:47:21  
5-SEP-1984 00:12:35

VAX/VMS Macro V04-00  
[DRIVER.SRC]DMDRIVER.MAR;1

Page 41  
(1)

UCBSW_DA	=	000000BC		
UCBSW_DC	=	000000BE		
UCBSW_DEVBUSIZ	=	00000042		
UCBSW_DEVSTS	=	00000068		
UCBSW_DM_AS		000000DC		
UCBSW_DM_BA		000000D2		
UCBSW_DM_CS1		000000CE		
UCBSW_DM_CS2		000000D6		
UCBSW_DM_DA		000000D4		
UCBSW_DM_DB		000000F4		
UCBSW_DM_DC		000000DE		
UCBSW_DM_DPN		000000E6		
UCBSW_DM_DS		000000D8		
UCBSW_DM_DTYP		000000CC		
UCBSW_DM_ER		000000DA		
UCBSW_DM_MR1		000000E0		
UCBSW_DM_MR2		000000E2		
UCBSW_DM_MR3		000000E4		
UCBSW_DM_WC		000000D0		
UCBSW_ECT	=	000000C4		
UCBSW_EC2	=	000000C6		
UCBSW_FUNC	=	0000009A		
UCBSW_OFFSET	=	000000C8		
UCBSW_STS	=	00000064		
UCBSW_UNIT	=	00000054		
UNLOAD		000001D0	R	03
VECSB_DATAPATH	=	00000013		
VECSL_IDB	=	00000008		
VECSL_INITIAL	=	0000000C		
VECSL_UNITINIT	=	00000018		
VECSS_DATAPATH	=	00000005		
VECSS_MAPREG	=	0000000F		
VECSV_DATAPATH	=	00000000		
VECSV_MAPREG	=	00000000		
VECSW_MAPREG	=	00000010		
WRITECHECK		000001E7	R	03
WRITEDATA		000001EE	R	03
WRITEHEAD		000001E7	R	03
XFER		000004E2	R	03

+-----+  
! Psect synopsis !  
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000100 ( 256.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
\$\$\$105_PROLOGUE	00000087 ( 135.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$\$\$115_DRIVER	00000A56 ( 2646.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

-----  
! Performance indicators !  
-----

Phase	Page faults	CPU Time	Elapsed Time
-----	-----	-----	-----
Initialization	34	00:00:00.07	00:00:01.40
Command processing	118	00:00:00.38	00:00:04.95
Pass 1	629	00:00:20.47	00:01:18.24
Symbol table sort	0	00:00:02.69	00:00:10.91
Pass 2	326	00:00:04.56	00:00:18.34
Symbol table output	47	00:00:00.26	00:00:01.50
Psect synopsis output	2	00:00:00.02	00:00:00.08
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1158	00:00:28.45	00:01:55.43

The working set limit was 2250 pages.  
164077 bytes (321 pages) of virtual memory were used to buffer the intermediate code.  
There were 130 pages of symbol table space allocated to hold 2395 non-local and 84 local symbols.  
1792 source lines were read in Pass 1, producing 23 object records in Pass 2.  
56 pages of virtual memory were used to define 53 macros.

-----  
! Macro library statistics !  
-----

Macro library name	Macros defined
-----	-----
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	37
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	10
TOTALS (all libraries)	47

2514 GETS were required to define 47 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:DMDRIVER/OBJ=OBJ\$:DMDRIVER MSRC\$:DMDRIVER/UPDATE=(ENH\$:DMDRIVER)+EXECML\$/LIB

0109 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

